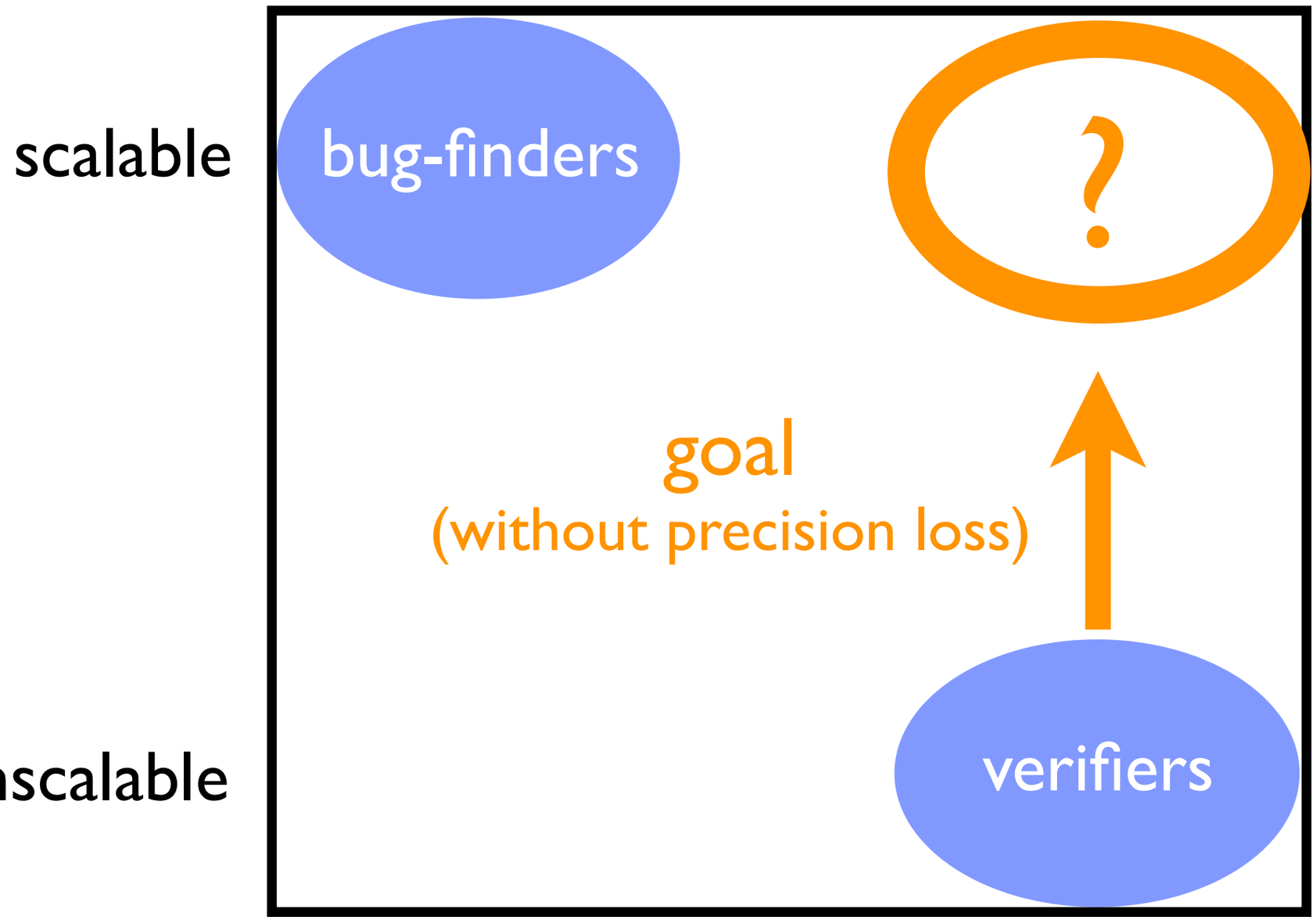
Spatial and Temporal Localization Techniques for Global Static Analysis

Hakjoo Oh
(with Kihong Heo, Wonchan Lee, Woosuk Lee, Kwangkeun Yi)

ROSAEC Center
Seoul National University

Oct. 26, 2012
AVDCPS 2012, Changsha, China

Dichotomy in Static Analysis



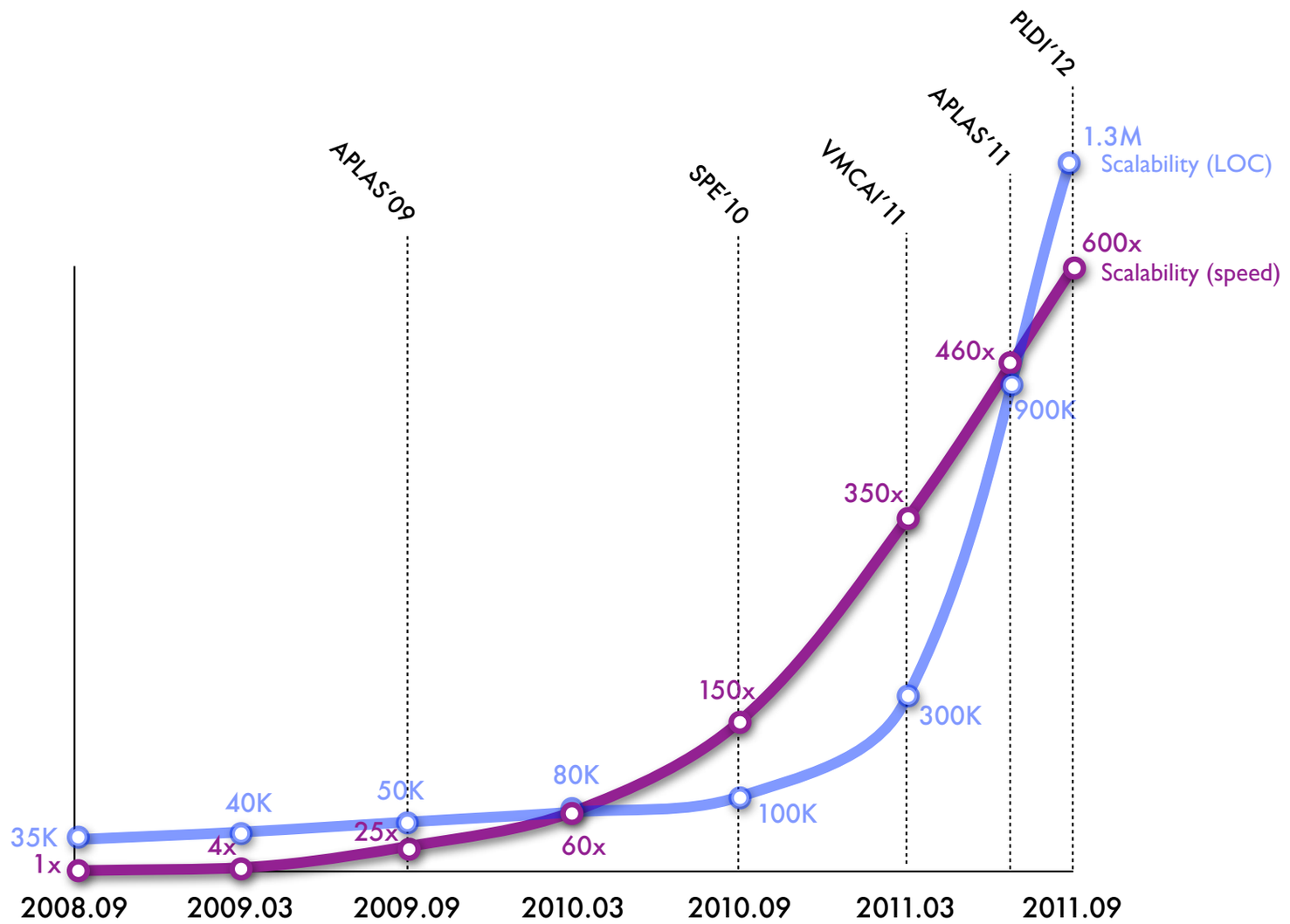
Our Story

- 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

ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 NRF ERC

This Talk

- How we achieved
  - sound design of Sparrow
  - spatial & temporal localizations

Design of Sparrow

The Early Bird

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 a next program point of c')

Commands

$$lv := e \mid lv := \mathsf{alloc}(a) \mid \mathsf{assume}(x < e) \mid \mathsf{call}(f_x, e) \mid \mathsf{return}_f$$

expression	e	$\rightarrow$	$n \mid e + e \mid lv \mid \&lv$
l-value	lv	$\rightarrow$	$x \mid *e \mid e[e] \mid e.x$
allocation	a	$\rightarrow$	$[e]_l \mid \{x\}_l$

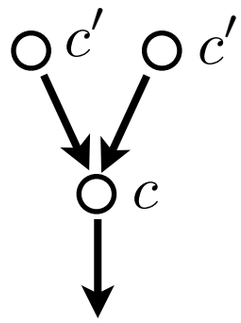
Abstract Semantics

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

$$
\begin{array}{rcl}
[\![\hat{P}]\!] \in \mathbb{C} \to \hat{\mathbb{S}} & = & fix\,\hat{F} \\
\hat{\mathbb{S}} & = & \hat{\mathbb{L}} \to \hat{\mathbb{V}}
\end{array}
\qquad
\begin{array}{rcl}
\hat{\mathbb{L}} & = & Var + AllocSite + AllocSite \times FieldName \\
\hat{\mathbb{V}} & = & \hat{\mathbb{Z}} \times 2^{\hat{\mathbb{L}}} \times 2^{AllocSite \times \hat{\mathbb{Z}} \times \hat{\mathbb{Z}}} \times 2^{AllocSite \times 2^{FieldName}} \\
\hat{\mathbb{Z}} & = & \{[l,u] \mid l,u \in \mathbb{Z} \cup \{-\infty,+\infty\} \wedge l \leq u\} \cup \{\bot\}
\end{array}
$$

- 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

Abstract Semantics

$$\hat{f}_c(\hat{s}) = \begin{cases} \hat{s}[\hat{\mathcal{L}}(lv)(\hat{s}) \overset{w}{\mapsto} \hat{\mathcal{V}}(e)(\hat{s})] & \mathsf{cmd}(c) = lv := e \\ \hat{s}[\hat{\mathcal{L}}(lv)(\hat{s}) \overset{w}{\mapsto} \langle \bot, \bot, \{\langle l, [0,0], \hat{\mathcal{V}}(e)(\hat{s}).1 \rangle\}, \bot \rangle] & \mathsf{cmd}(c) = lv := \mathsf{alloc}([e]_l) \\ \hat{s}[\hat{\mathcal{L}}(lv)(\hat{s}) \overset{w}{\mapsto} \langle \bot, \bot, \bot, \{\langle l, \{x\} \rangle\} \rangle] & \mathsf{cmd}(c) = lv := \mathsf{alloc}(\{x\}_l) \\ \hat{s}[x \mapsto \langle \hat{s}(x).1 \sqcap [-\infty, \mathsf{u}(\hat{\mathcal{V}}(e)(\hat{s}).1)], \hat{s}(x).2, \hat{s}(x).3, \hat{s}(x).4 \rangle] & \mathsf{cmd}(c) = \mathsf{assume}(x < e) \\ \hat{s}[x \mapsto \hat{\mathcal{V}}(e)(\hat{s})] & \mathsf{cmd}(c) = \mathsf{call}(f_x, e) \\ \hat{s} & \mathsf{cmd}(c) = \mathsf{return}_f \end{cases}$$

$$\begin{array}{rcl} \hat{\mathcal{V}}(e) & \in & \hat{\mathbb{S}} \to \hat{\mathbb{V}} \\ \hat{\mathcal{V}}(n)(\hat{s}) & = & \langle [n,n], \bot, \bot, \bot \rangle \\ \hat{\mathcal{V}}(e_1 \ + \ e_2)(\hat{s}) & = & \hat{\mathcal{V}}(e_1)(\hat{s}) \hat{+} \hat{\mathcal{V}}(e_2)(\hat{s}) \\ \hat{\mathcal{V}}(lv)(\hat{s}) & = & \bigsqcup \{\hat{s}(a) \mid a \in \hat{\mathcal{L}}(lv)(\hat{s})\} \\ \hat{\mathcal{V}}(\&lv)(\hat{s}) & = & \langle \bot, \hat{\mathcal{L}}(lv)(\hat{s}), \bot, \bot \rangle \end{array}$$

$$\begin{array}{rcl} \hat{\mathcal{L}}(lv) & \in & \hat{\mathbb{S}} \to 2^{\hat{\mathbb{L}}} \\ \hat{\mathcal{L}}(x)(\hat{s}) & = & \{x\} \\ \hat{\mathcal{L}}(*e)(\hat{s}) & = & \hat{\mathcal{V}}(e)(\hat{s}).2 \cup \{l \mid \langle l, o, s \rangle \in \hat{\mathcal{V}}(e)(\hat{s}).3\} \\ & & \cup \{\langle l, x \rangle \mid \langle l, \{x\} \rangle \in \hat{\mathcal{V}}(e)(\hat{s}).4\} \\ \hat{\mathcal{L}}(e_1[e_2])(\hat{s}) & = & \{l \mid \langle l, o, s \rangle \in \hat{\mathcal{V}}(e_1)(\hat{s}).3\} \\ \hat{\mathcal{L}}(e.x)(\hat{s}) & = & \{\langle l, x \rangle \mid \langle l, \{x\} \rangle \in \hat{\mathcal{V}}(e)(\hat{s}).4\} \end{array}$$

Computing $\bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\bot})$

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

$\hat{X}, \hat{X}' \in \mathbb{C} \to \hat{\mathbb{S}}$
$\hat{f}_c \in \hat{\mathbb{S}} \to \hat{\mathbb{S}}$

$\hat{X} := \hat{X}' := \lambda c. \bot$
repeat
  $\hat{X}' := \hat{X}$
  for all $c \in \mathbb{C}$ **do**
    $\hat{X}(c) := \hat{f}_c(\bigsqcup_{c' \hookrightarrow c} X(c'))$
until $\hat{X} \sqsubseteq \hat{X}'$

Naive fixpoint algorithm

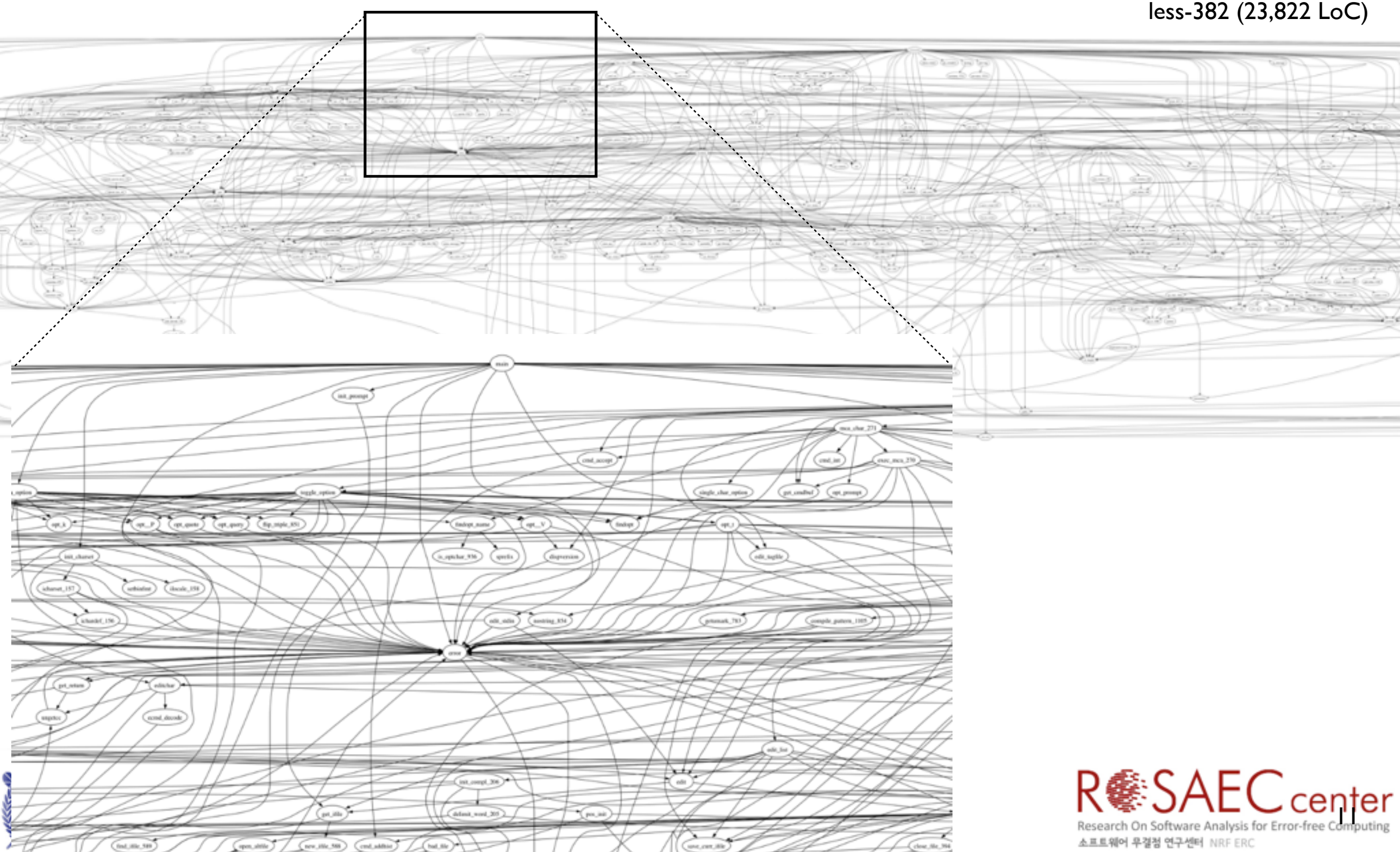
$W \in \mathit{Worklist} = 2^{\mathbb{C}}$
$\hat{X} \in \mathbb{C} \to \hat{\mathbb{S}}$
$\hat{f}_c \in \hat{\mathbb{S}} \to \hat{\mathbb{S}}$

$W := \mathbb{C}$
$\hat{X} := \lambda c. \bot$
repeat
  $c := \mathsf{choose}(W)$
  $\hat{s} := \hat{f}_c(\bigsqcup_{c' \hookrightarrow c} X(c'))$
  if $\hat{s} \not\sqsubseteq \hat{X}(c)$
    $W := W \cup \{c' \in \mathbb{C} \mid c \hookrightarrow c'\}$
    $\hat{X}(c) := \hat{X}(c) \sqcup \hat{s}$
until $W = \emptyset$

Worklist algorithm

Direct Implementation (convention) Too Weak To Scale

less-382 (23,822 LoC)



Improving Scalability

Key Idea: Localization

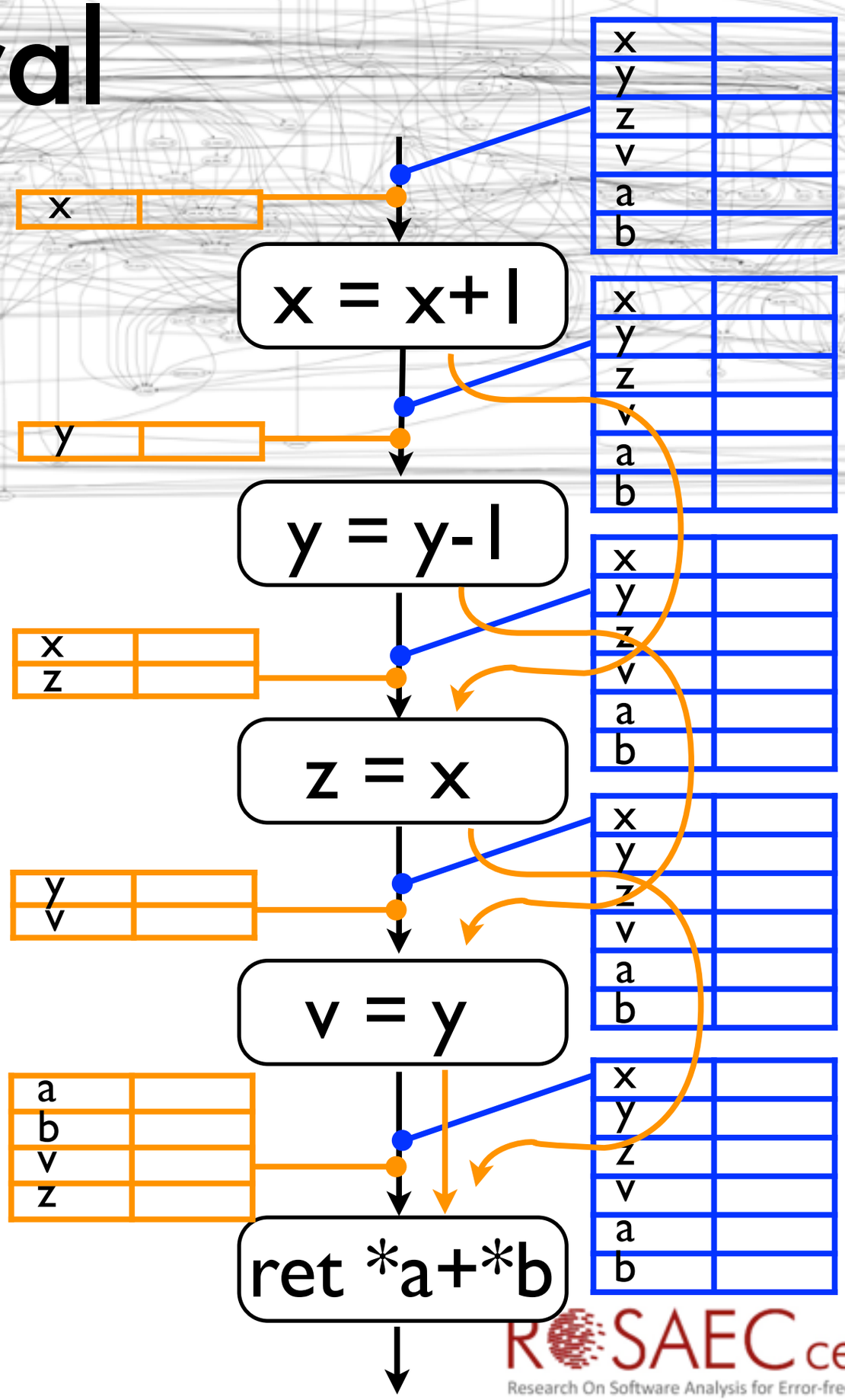
("framing" in separation logic)

"Right Part at Right Moment"

- Spatial localization [VMCAI'11,APLAS'11]
- Temporal localization [PLDI'12]

Spatial & Temporal Localizations

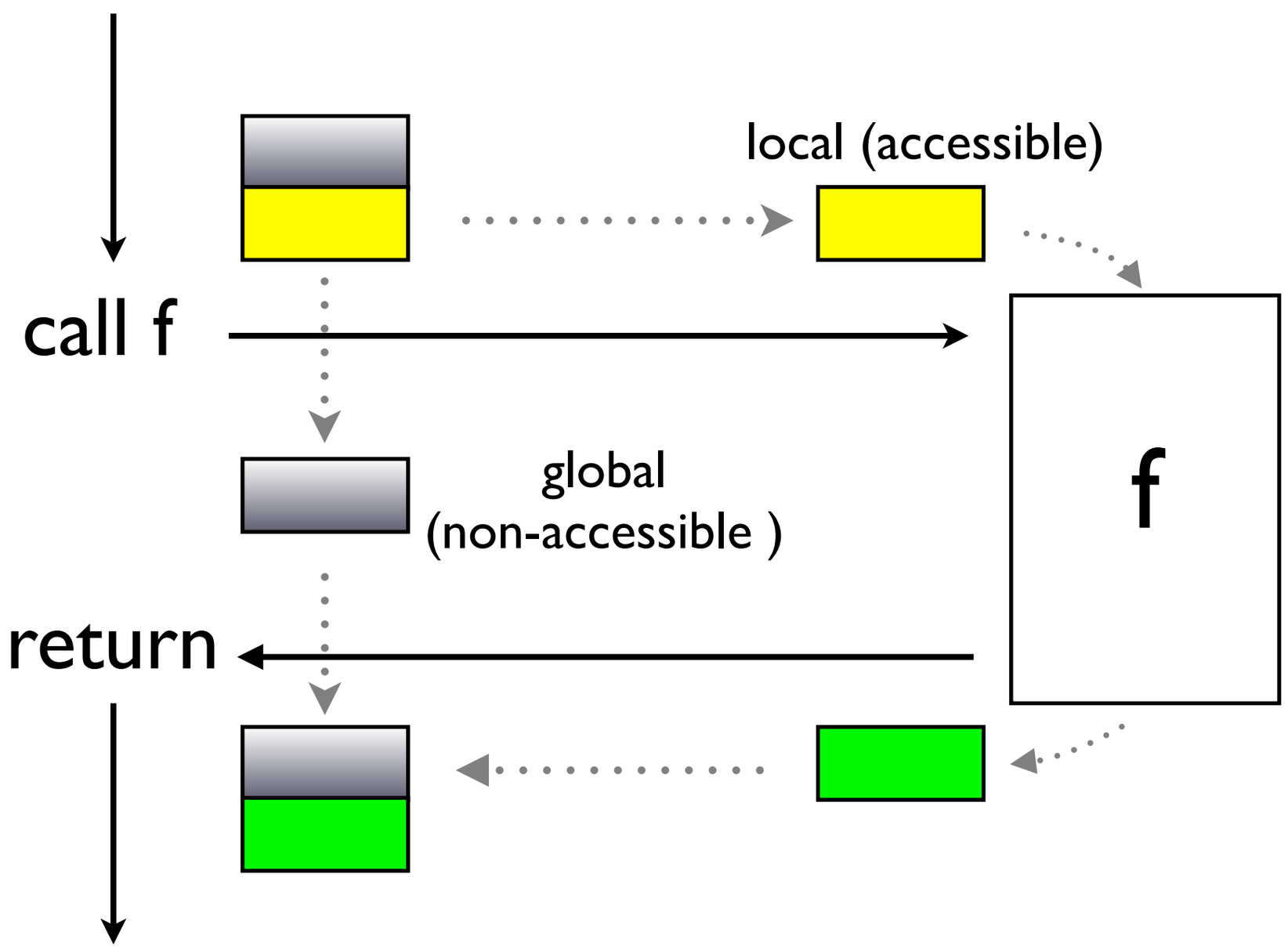
$$\hat{X}, \hat{X}' \in \mathbb{C} \to \hat{\mathbb{S}}$$
$$\hat{f}_c \in \hat{\mathbb{S}} \to \hat{\mathbb{S}}$$
$$\hat{X} := \hat{X}' := \lambda c.\bot$$
repeat
$$\quad \hat{X}' := \hat{X}$$
$$\quad \textbf{for all } c \in \mathbb{C} \textbf{ do}$$
$$\quad\quad \hat{X}(c) := \hat{f}_c(\bigsqcup_{c' \hookrightarrow c} \hat{X}(c'))$$
until $\hat{X} \sqsubseteq \hat{X}'$



Spatial Localization

Spatial Localization

(Memory localization)



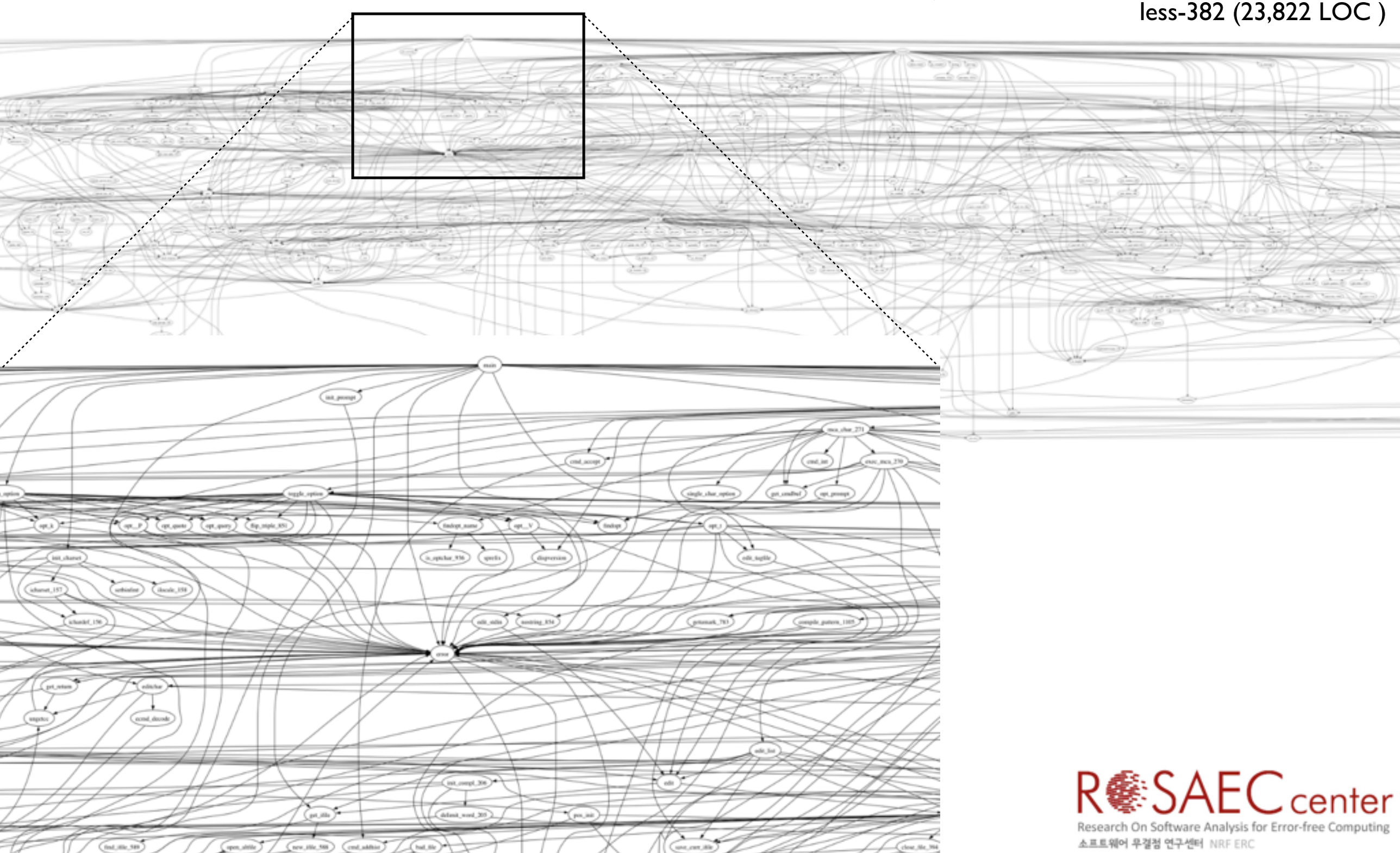
Benefits of Localization

```
int g;

int f() {...}

int main() {
   g = 0;  f();
   g = 1;  f();
}
```

f does not access g

Localization Is Vital

less-382 (23,822 LOC)



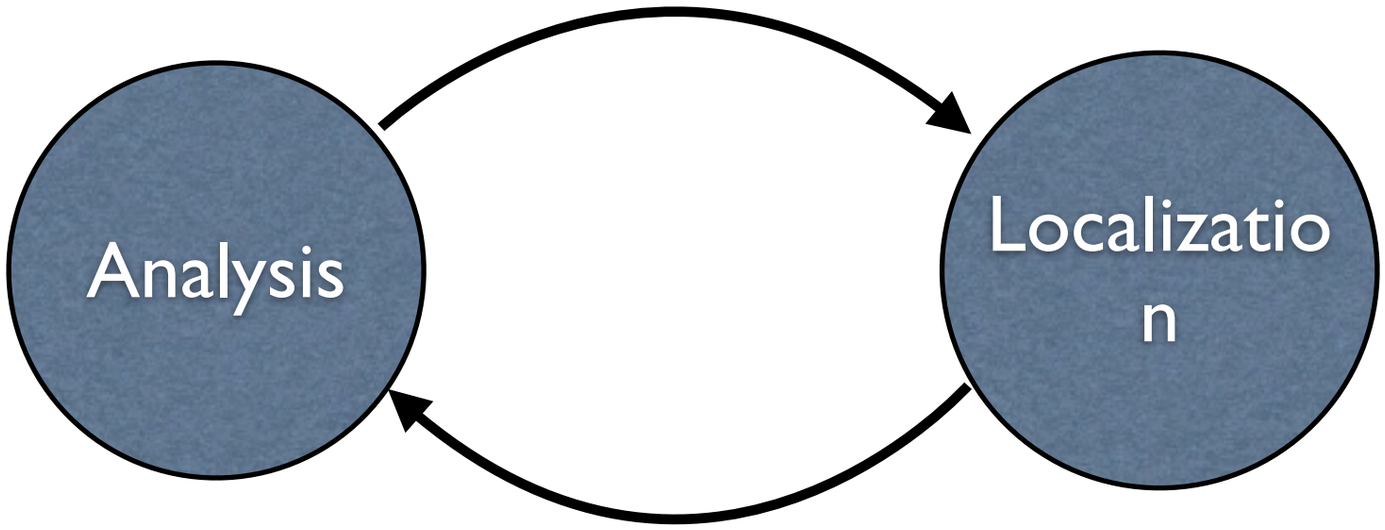
Localization Is Vital

less-382 (23,822 LOC)

On average 755 re-analyses per procedure

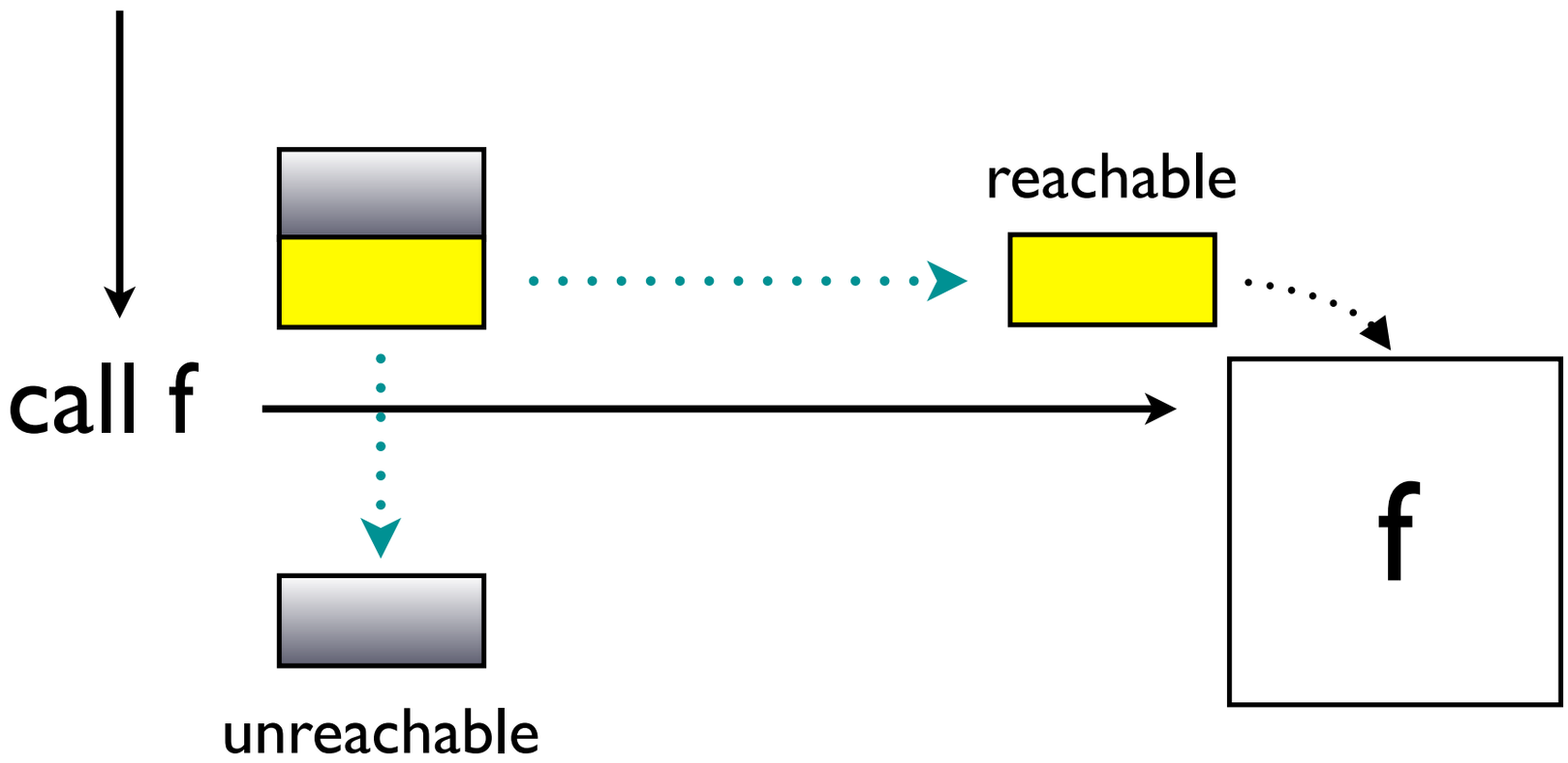
The Catch-22 Situation

The optimal localization is impossible



Reachability-based Localization
(abstract garbage collection)

- Remove the unreachable from params and globals



Too Conservative in Practice

Program	LOC	accessed memory / reachable memory
spell-1.0	2,213	5 / 453 (1.1%)
barcode-0.96	4,460	19 / 1175 (1.6%)
httptunnel-3.3	6,174	10 / 673 (1.5%)
gzip-1.2.4a	7,327	22 / 1002 (2.2%)
jwhois-3.0.1	9,344	28 / 830 (3.4%)
parser	10,900	75 / 1787 (4.2%)
bc-1.06	13,093	24 / 824 (2.9%)
less-290	18,449	86 / 1546 (5.6%)

average : 4%

VMCAI'11

Access-based Localization

pre-analysis —(conservative access information)→ actual analysis

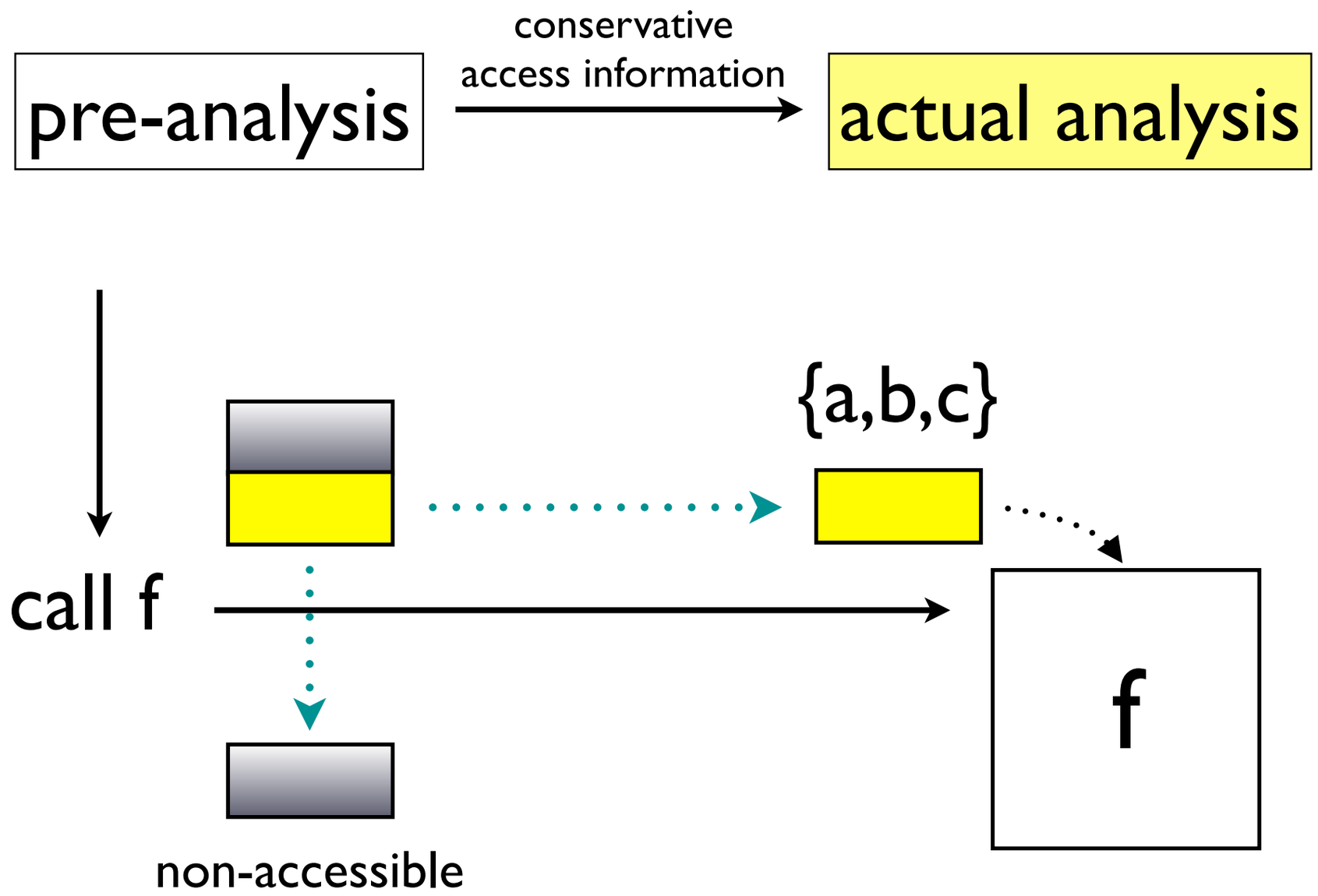
Over-approximation of actual access info. {a,b,c}

U

actual access info. {a,b}

f

Access-based Localization



Performance of sound & global

Programs	LOC	$\text{Interval}_{\text{vanilla}}$		$\text{Interval}_{\text{base}}$		Spd↑1	Mem↓1	$\text{Interval}_{\text{sparse}}$						Spd↑2	Mem↓2
		Time	Mem	Time	Mem			Dep	Fix	Total	Mem	$\hat{D}(c)$	$\hat{U}(c)$		
gzip-1.2.4a	7K	772	240	14	65	55 x	73 %	2	1	3	63	2.4	2.5	5 x	3 %
bc-1.06	13K	1,270	276	96	126	13 x	54 %	4	3	7	75	4.6	4.9	14 x	40 %
tar-1.13	20K	12,947	881	338	177	38 x	80 %	6	2	8	93	2.9	2.9	42 x	47 %
less-382	23K	9,561	1,113	1,211	378	8 x	66 %	27	6	33	127	11.9	11.9	37 x	66 %
make-3.76.1	27K	24,240	1,391	1,893	443	13 x	68 %	16	5	21	114	5.8	5.8	90 x	74 %
wget-1.9	35K	44,092	2,546	1,214	378	36 x	85 %	8	3	11	85	2.4	2.4	110 x	78 %
screen-4.0.2	45K	∞	N/A	31,324	3,996	N/A	N/A	724	43	767	303	53.0	54.0	41 x	92 %
a2ps-4.14	64K	∞	N/A	3,200	1,392	N/A	N/A	31	9	40	353	2.6	2.8	80 x	75 %
bash-2.05a	105K	∞	N/A	1,683	1,386	N/A	N/A	45	22	67	220	3.0	3.0	25 x	84 %
lsh-2.0.4	111K	∞	N/A	45,522	5,266	N/A	N/A	391	80	471	577	21.1	21.2	97 x	89 %
sendmail-8.13.6	130K	∞	N/A	∞	N/A	N/A	N/A	517	227	744	678	20.7	20.7	N/A	N/A
nethack-3.3.0	211K	∞	N/A	∞	N/A	N/A	N/A	14,126	2,247	16,373	5,298	72.4	72.4	N/A	N/A
vim60	227K	∞	N/A	∞	N/A	N/A	N/A	17,518	6,280	23,798	5,190	180.2	180.3	N/A	N/A
emacs-22.1	399K	∞	N/A	∞	N/A	N/A	N/A	29,552	8,278	37,830	7,795	285.3	285.5	N/A	N/A
python-2.5.1	435K	∞	N/A	∞	N/A	N/A	N/A	9,677	1,362	11,039	5,535	108.1	108.1	N/A	N/A
linux-3.0	710K	∞	N/A	∞	N/A	N/A	N/A	26,669	6,949	33,618	20,529	76.2	74.8	N/A	N/A
gimp-2.6	959K	∞	N/A	∞	N/A	N/A	N/A	3,751	123	3,874	3,602	4.1	3.9	N/A	N/A
ghostscript-9.00	1,363K	∞	N/A	∞	N/A	N/A	N/A	14,116	698	14,814	6,384	9.7	9.7	N/A	N/A

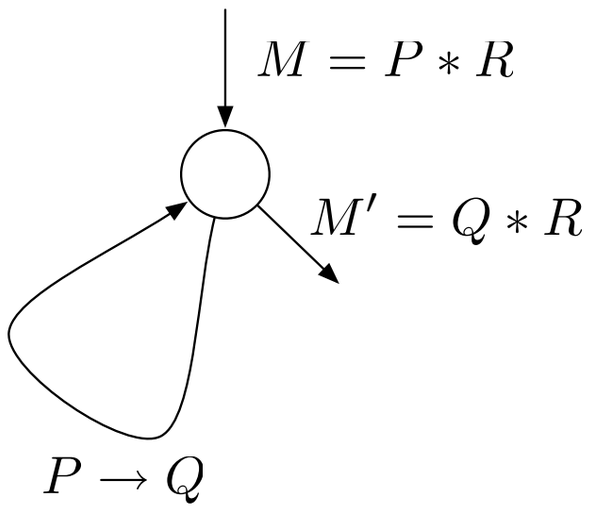
reachability

access-based localization

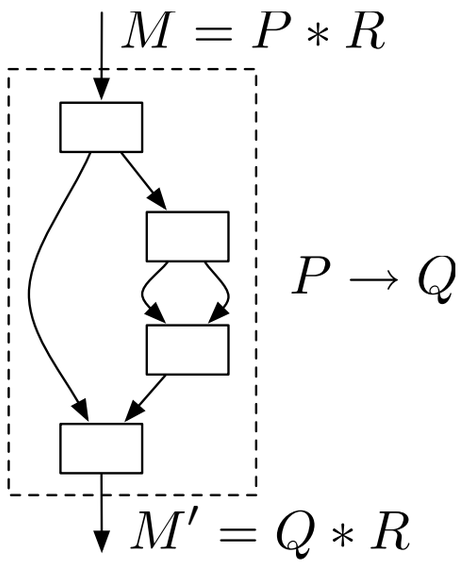
Block-level Localization

Access-based localization at any level

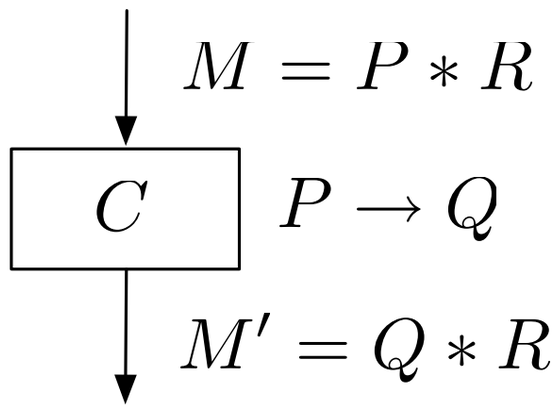
loops



branches

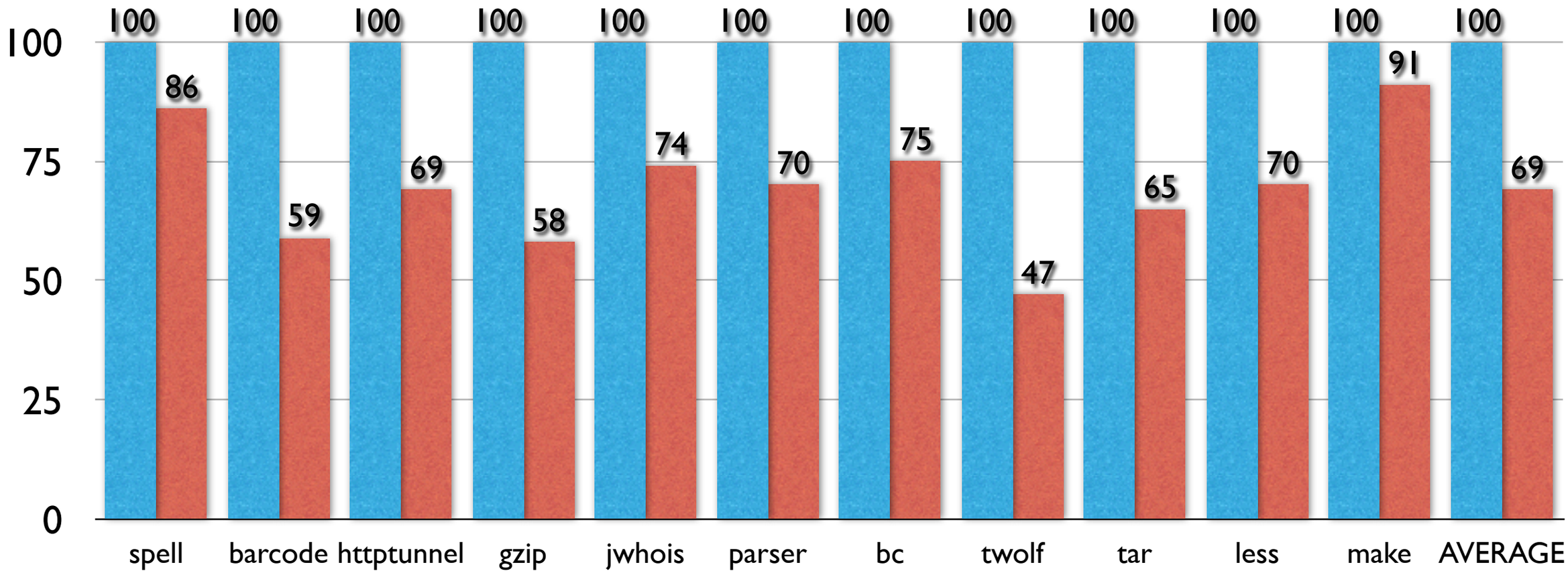


basic blocks



Performance

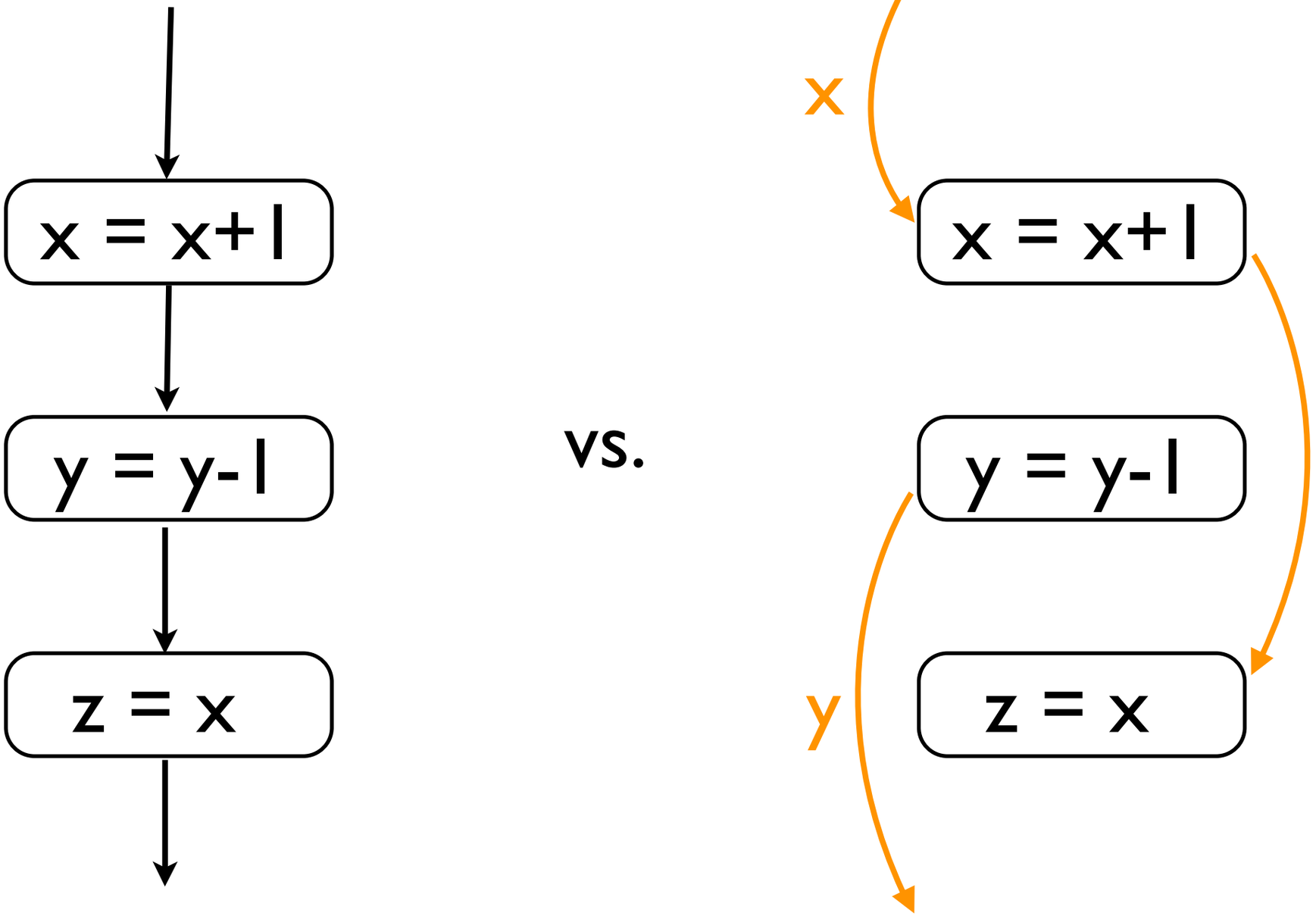
On average 31% reduction in time (k=6)



Temporal Localization

(and spatial localization automatically follows)

Temporal Localization
(sparse analysis)



PLDI'12

Precision Preserving Sparse Analysis Framework

$$\hat{F} : \hat{D} \to \hat{D} \quad \xRightarrow{\text{sparsify}} \quad \hat{F}_s : \hat{D} \to \hat{D}$$

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

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

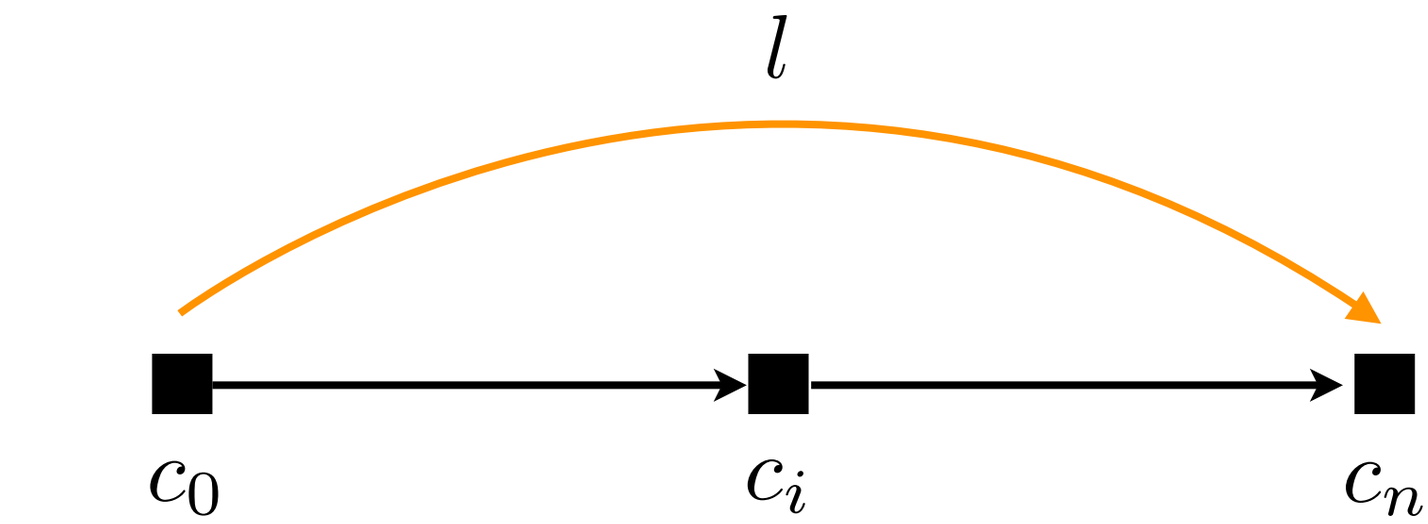
$$\hat{F}_a(\hat{X}) = \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \overset{l}{\rightsquigarrow}_a c} \hat{X}(c')|_l).$$

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

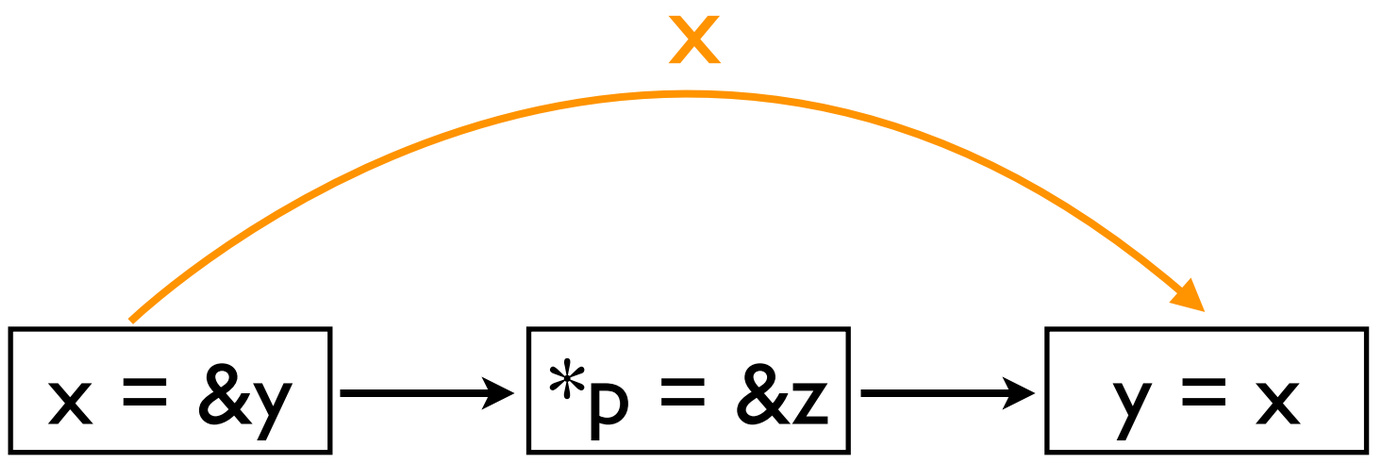
$$\mathsf{D}(c) \triangleq \{l \in \hat{\mathbb{L}} \mid \exists \hat{s} \sqsubseteq \bigsqcup_{c' \hookrightarrow c} (fix\hat{F})(c').\hat{f}_c(\hat{s})(l) \neq \hat{s}(l)\}.$$

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

Preserving

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

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

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

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 $\hat{D}$ & $\hat{U}$

- over-approximation

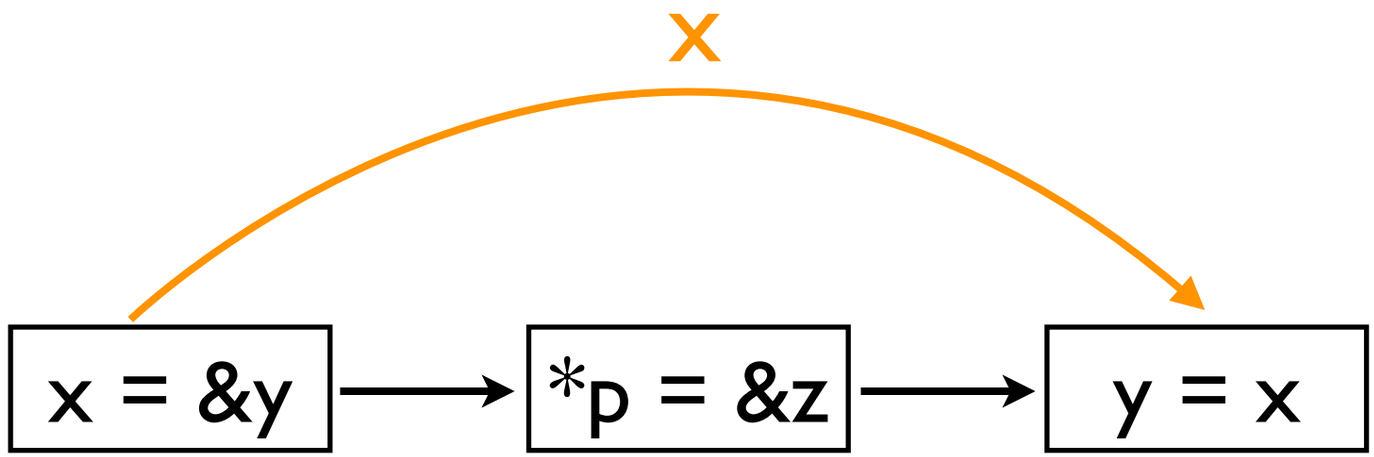
$$\mathsf{D}(c) \supseteq \mathsf{D}(c) \wedge \mathsf{U}(c) \supseteq \mathsf{U}(c)$$

- spurious definitions should be also included in uses

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

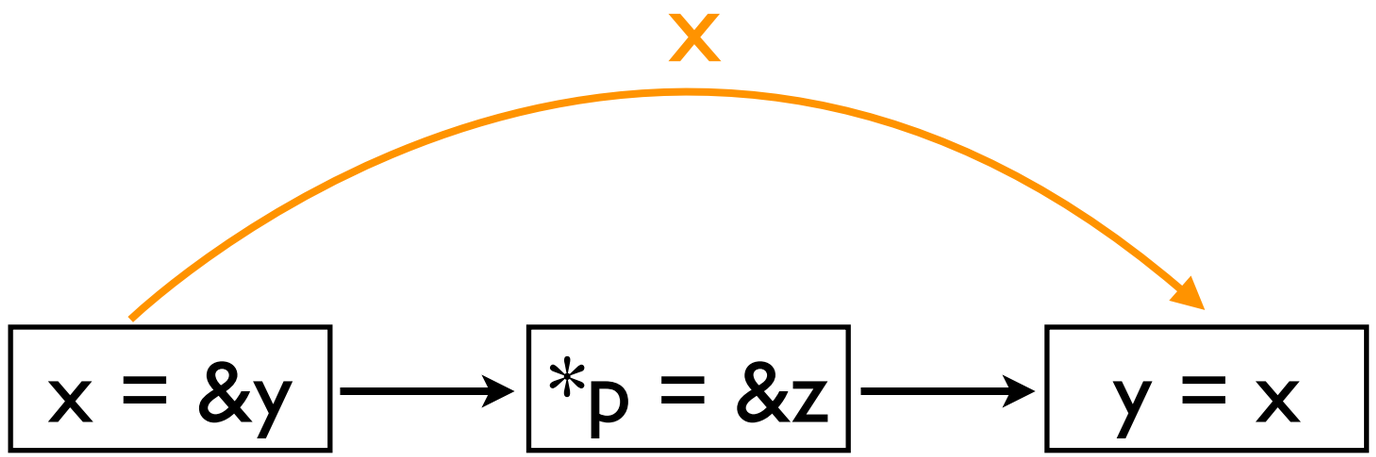
spurious definitions

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



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

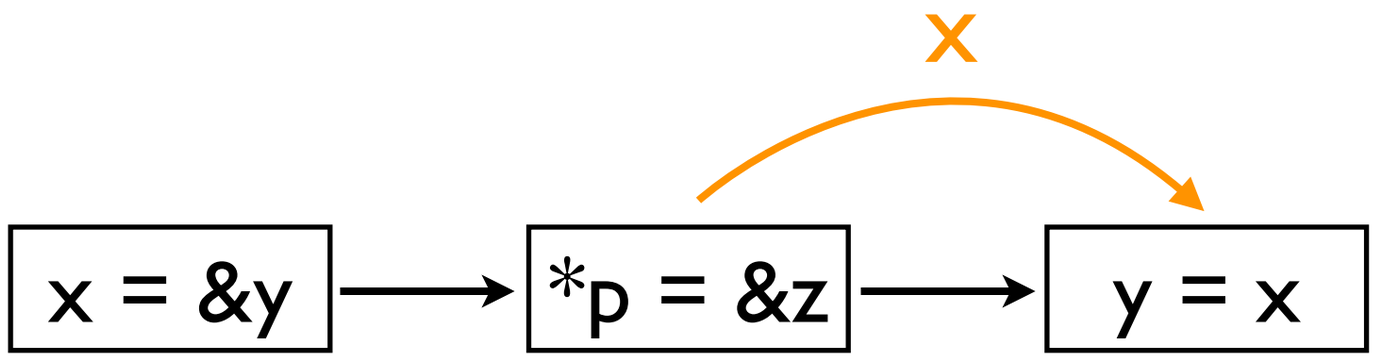
Why the Conditions of $\hat{D}$ & $\hat{U}$



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

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

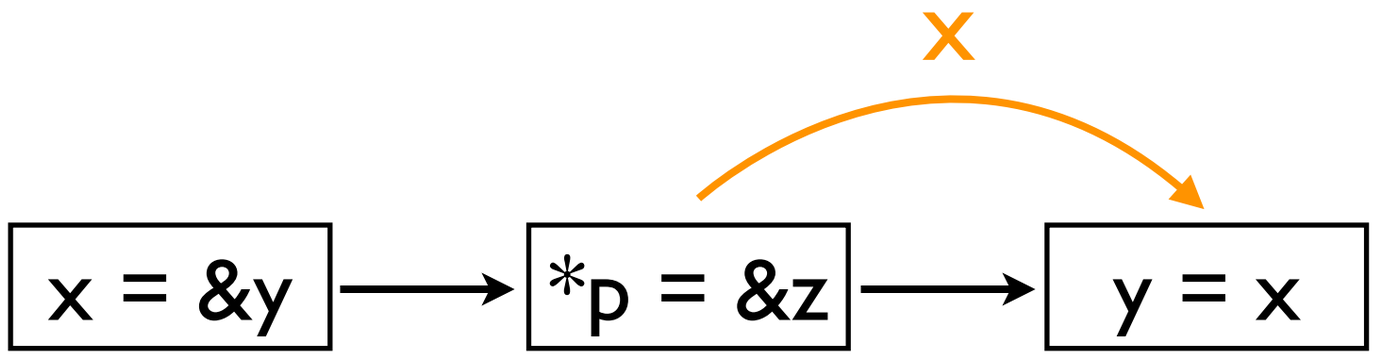
Why the Conditions of $\hat{D}$ & $\hat{U}$



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

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

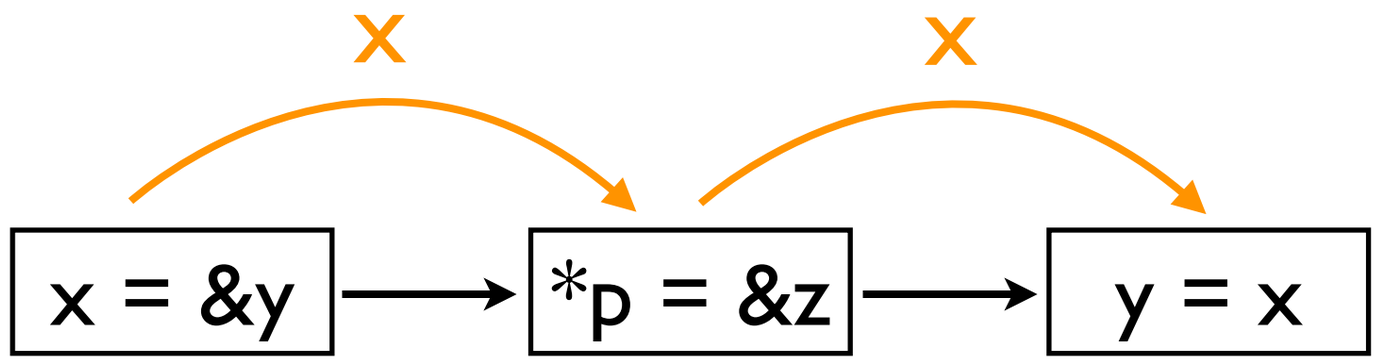
Why the Conditions of $\hat{D}$ & $\hat{U}$



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

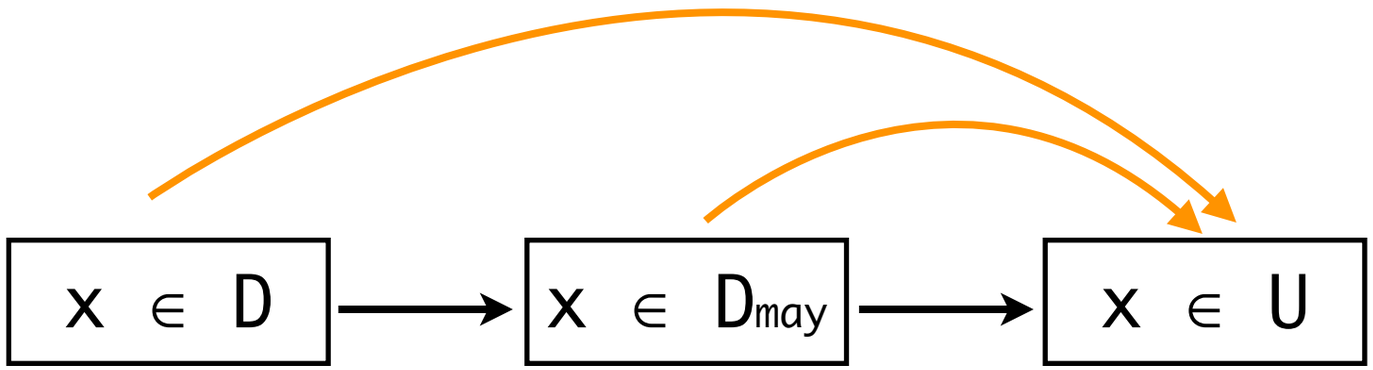
$$\mathbb{C} \to \hat{\mathbb{S}} \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{\mathbb{S}} \qquad \hat{F}_p = \lambda \hat{s}.(\bigsqcup_{c \in \mathbb{C}} \hat{f}_c(\hat{s}))$$

- 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)

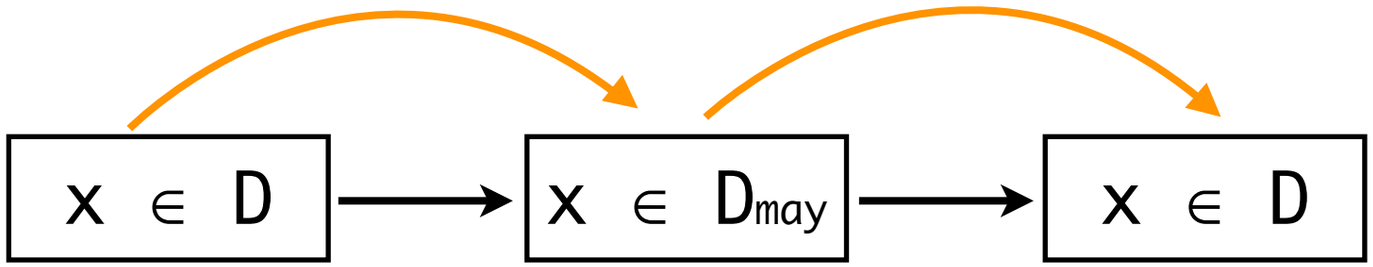
- Different notion of data dependency



def-use chains fail to preserve original precision

Existing Sparse Techniques
(developed mostly in dfa community)

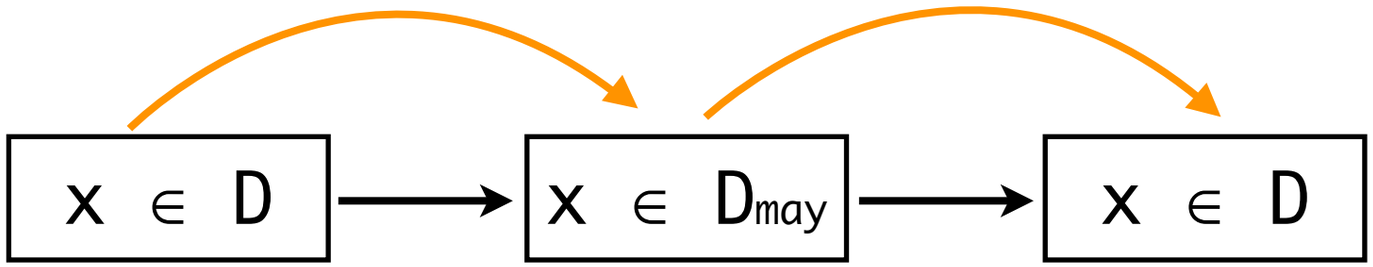
- Different notion of data dependency



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

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

Programs	LOC	$\text{Interval}_{\text{vanilla}}$		$\text{Interval}_{\text{base}}$		Spd↑$_1$	Mem↓$_1$	$\text{Interval}_{\text{sparse}}$						Spd↑$_2$	Mem↓$_2$
		Time	Mem	Time	Mem			Dep	Fix	Total	Mem	$\hat{D}(c)$	$\hat{U}(c)$		
gzip-1.2.4a	7K	772	240	14	65	55 x	73 %	2	1	3	63	2.4	2.5	5 x	3 %
bc-1.06	13K	1,270	276	96	126	13 x	54 %	4	3	7	75	4.6	4.9	14 x	40 %
tar-1.13	20K	12,947	881	338	177	38 x	80 %	6	2	8	93	2.9	2.9	42 x	47 %
less-382	23K	9,561	1,113	1,211	378	8 x	66 %	27	6	33	127	11.9	11.9	37 x	66 %
make-3.76.1	27K	24,240	1,391	1,893	443	13 x	68 %	16	5	21	114	5.8	5.8	90 x	74 %
wget-1.9	35K	44,092	2,546	1,214	378	36 x	85 %	8	3	11	85	2.4	2.4	110 x	78 %
screen-4.0.2	45K	∞	N/A	31,324	3,996	N/A	N/A	724	43	767	303	53.0	54.0	41 x	92 %
a2ps-4.14	64K	∞	N/A	3,200	1,392	N/A	N/A	31	9	40	353	2.6	2.8	80 x	75 %
bash-2.05a	105K	∞	N/A	1,683	1,386	N/A	N/A	45	22	67	220	3.0	3.0	25 x	84 %
lsh-2.0.4	111K	∞	N/A	45,522	5,266	N/A	N/A	391	80	471	577	21.1	21.2	97 x	89 %
sendmail-8.13.6	130K	∞	N/A	∞	N/A	N/A	N/A	517	227	744	678	20.7	20.7	N/A	N/A
nethack-3.3.0	211K	∞	N/A	∞	N/A	N/A	N/A	14,126	2,247	16,373	5,298	72.4	72.4	N/A	N/A
vim60	227K	∞	N/A	∞	N/A	N/A	N/A	17,518	6,280	23,798	5,190	180.2	180.3	N/A	N/A
emacs-22.1	399K	∞	N/A	∞	N/A	N/A	N/A	29,552	8,278	37,830	7,795	285.3	285.5	N/A	N/A
python-2.5.1	435K	∞	N/A	∞	N/A	N/A	N/A	9,677	1,362	11,039	5,535	108.1	108.1	N/A	N/A
linux-3.0	710K	∞	N/A	∞	N/A	N/A	N/A	26,669	6,949	33,618	20,529	76.2	74.8	N/A	N/A
gimp-2.6	959K	∞	N/A	∞	N/A	N/A	N/A	3,751	123	3,874	3,602	4.1	3.9	N/A	N/A
ghostscript-9.00	1,363K	∞	N/A	∞	N/A	N/A	N/A	14,116	698	14,814	6,384	9.7	9.7	N/A	N/A

spatial localization

spatial+temporal localization

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

For precise, sound, and scalable static analysis

- Define a global safe abstract interpreter
- Apply spatial & temporal localizations
- Resulting analysis scales with the same precision

Thank you