Scalable Global Static Analysis, Automation, and Secrecy

Kwangkeun Yi
Seoul National University, Korea

10/16/2018@MEMOCODE, Beijing

co-work with
Hakjoo Oh, Kihong Heo,
Wonchan Lee, Woosuk Lee,
Sungkeun Cho, Jeehoon Kang

PLDI(2014, 2012), TOPLAS(2015, 2014), SAS(2015), ICSE(2017), VMCAI(2012),...

Message

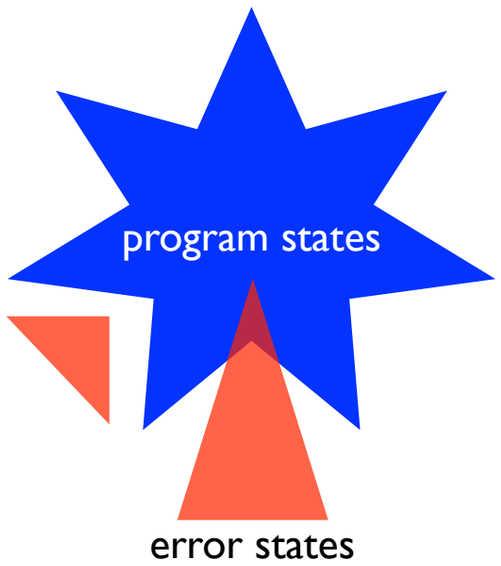
- Practical: Scalable, Sound, Precise, Global Static Analyses
  - General Sparse Analysis Framework for C-like Languages
- Matured: Automation (scalable anlyzr + verified validtr)
  - ZooBerry System
- Possible: Analysis in Secrecy

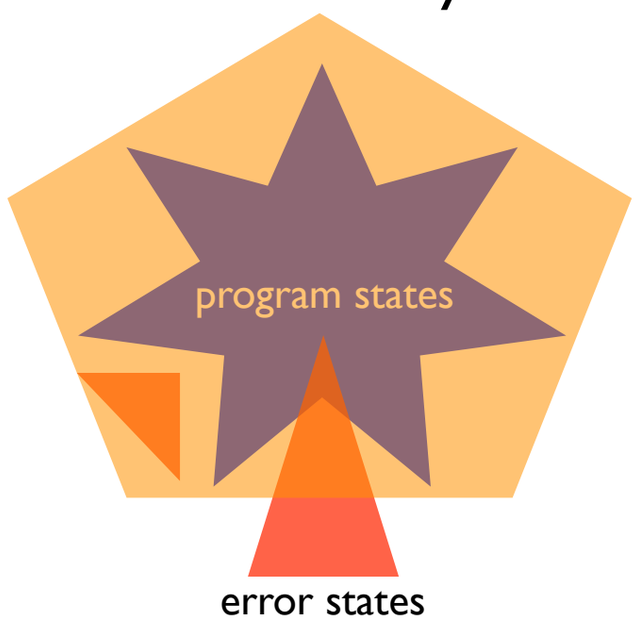
Static Analysis: Sound, Unsound, Useful

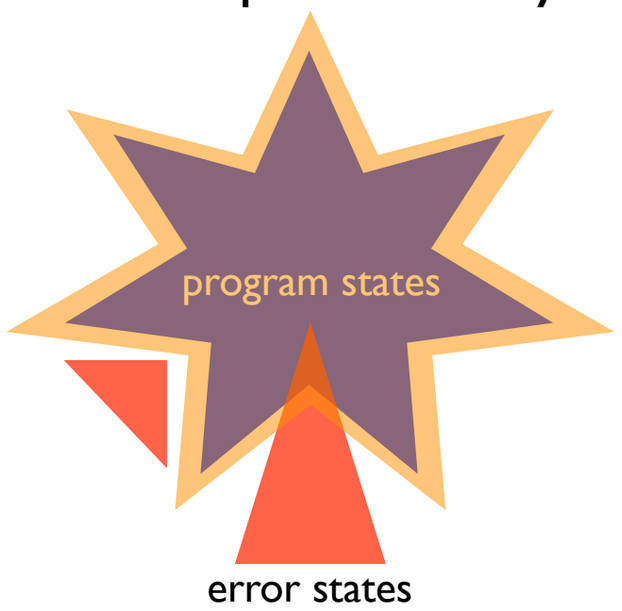
Automatic sound estimation of sw behaviors before execution

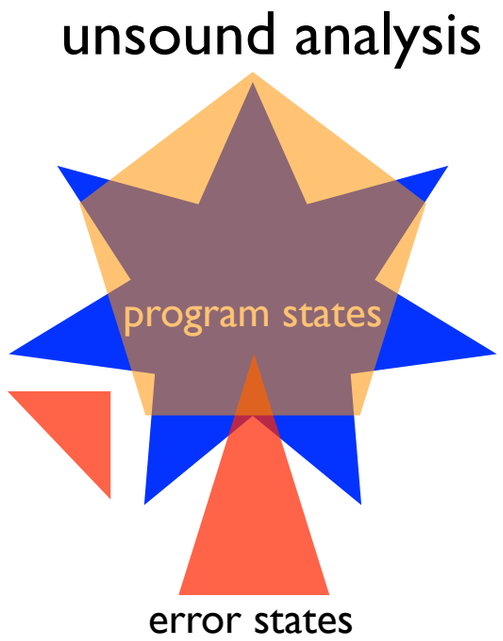
- under many names

theory	"abstract interpretation"
pl, se, veri.	"type system", "model checking", "theorem proving"
cmplr	"data-flow analysis", etc.









Static Analysis Example: Abstract Equations

```
        x = readInt;
1:
        while (x ≤ 99)
2:
                x++;
3:
        end
4:
```

Capture the dynamics by abstract equations; solve; reason.

$$
\begin{aligned}
x_1 &= [-\infty, +\infty] \; or \; x_3 \\
x_2 &= x_1 \; and \; [-\infty, 99] \\
x_3 &= x_2 \dot{+} 1 \\
x_4 &= x_1 \; and \; [100, +\infty]
\end{aligned}
$$

How to Design Sound Static Analyses?

Abstract Interpretation [CousotCousot]: a powerful design theory

- How to derive correct yet arbitrarily precise equations?
  - Non-obvious: ptrs, heap, exns, high-order ftns, etc.

```
x = readInt;
while (x ≤ 99)
      x++;
end
```

how? ⟹

$$\begin{array}{lcl} x_1 & = & [-\infty, +\infty] \ or \ x_3 \\ x_2 & = & x_1 \ and \ [-\infty, 99] \\ x_3 & = & x_2 \dot{+} 1 \\ x_4 & = & x_1 \ and \ [100, +\infty] \end{array}$$

  - Define an abstract semantics function $\hat{F}$ s.t. $\cdots$

- How to solve the equations in a finite time?

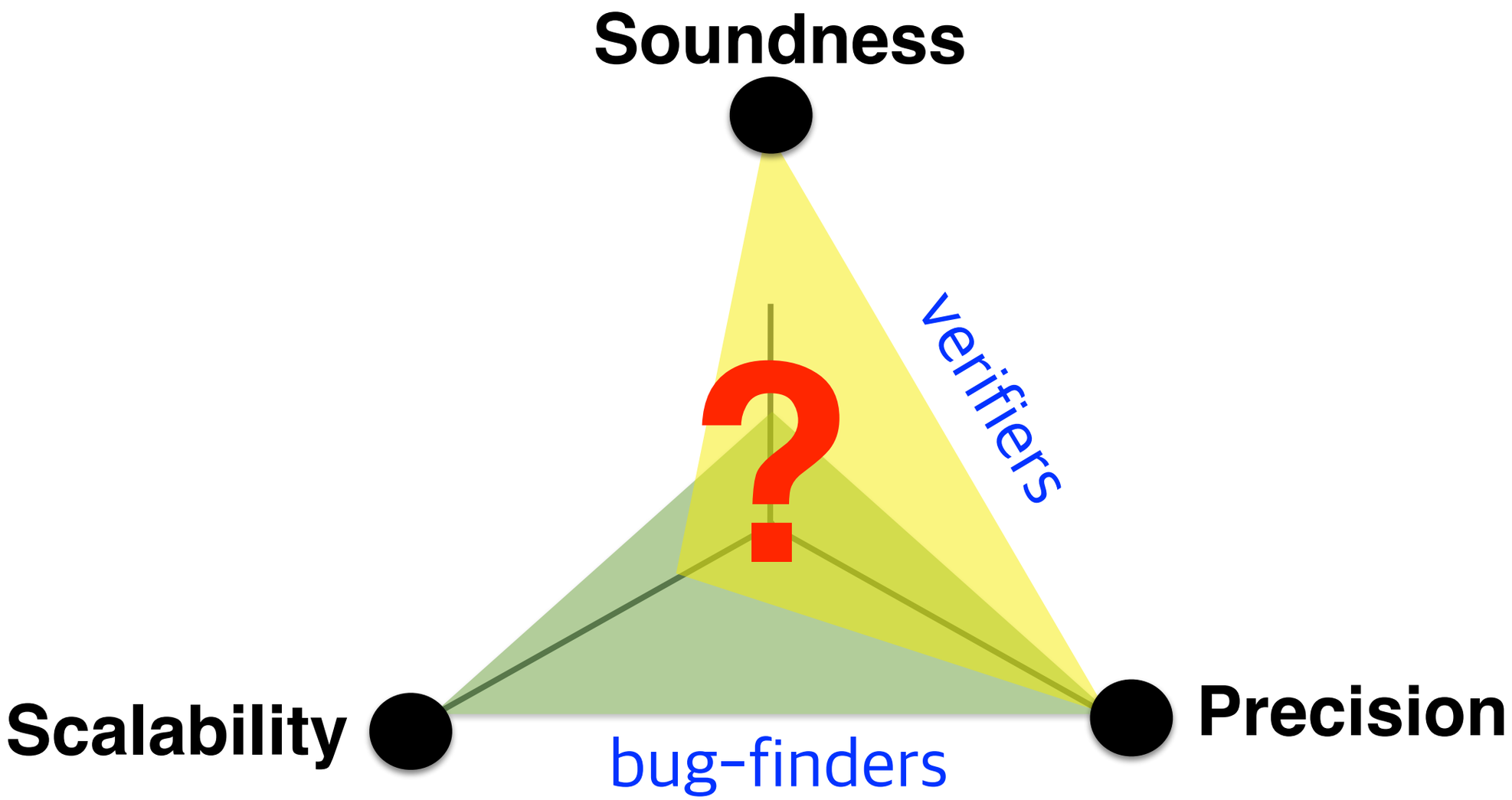
$$\begin{array}{lcl} x_1 & = & [-\infty, +\infty] \ or \ x_3 \\ x_2 & = & x_1 \ and \ [-\infty, 99] \\ x_3 & = & x_2 \dot{+} 1 \\ x_4 & = & x_1 \ and \ [100, \infty] \end{array}$$

how? ⟹

$$\begin{array}{lcl} x_1 & = & [-\infty, +\infty] \\ x_2 & = & [-\infty, 99] \\ x_3 & = & [-\infty, 100] \\ x_4 & = & [100, +\infty] \end{array}$$

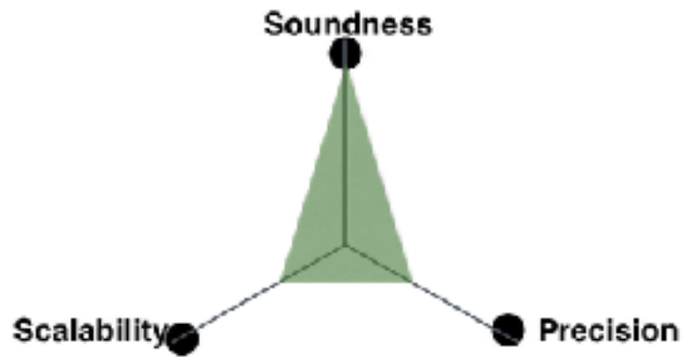
  - Fixpoint iterations for an upperbound of $fix\hat{F}$

Challenge in Static Analysis



Our Story

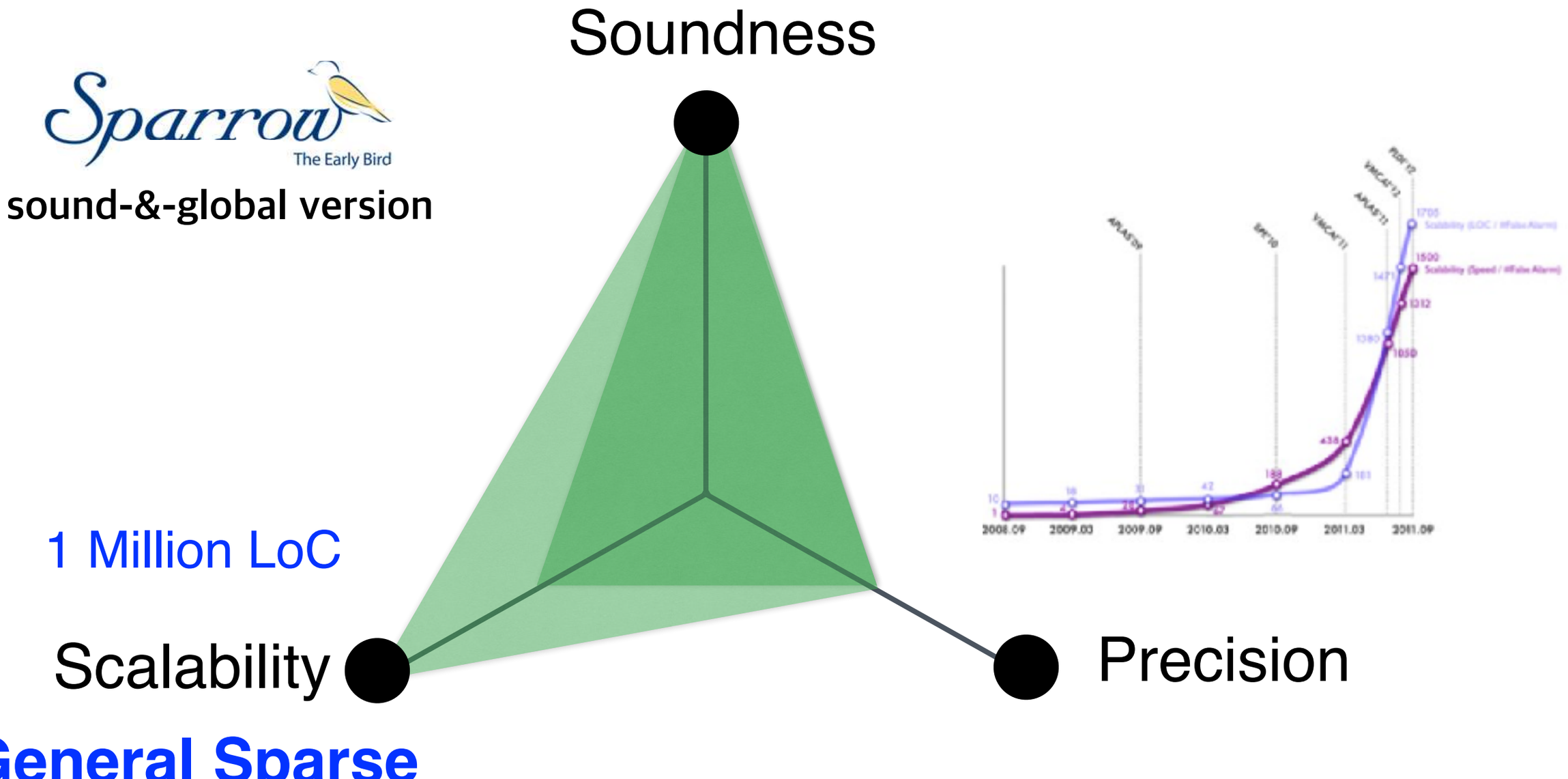
- In 2007-9, we commercialized Sparrow (The Early Bird)
  - memory-bug-finding tool for full C
  - designed in abstract interpretation framework
  - sound in design, unsound yet scalable in reality (non-global)
- Realistic workbench available
  - "let's try to achieve sound, precise, yet scalable global version"



Our Story: scalability

Sparrow
The Early Bird

sound-&-global version

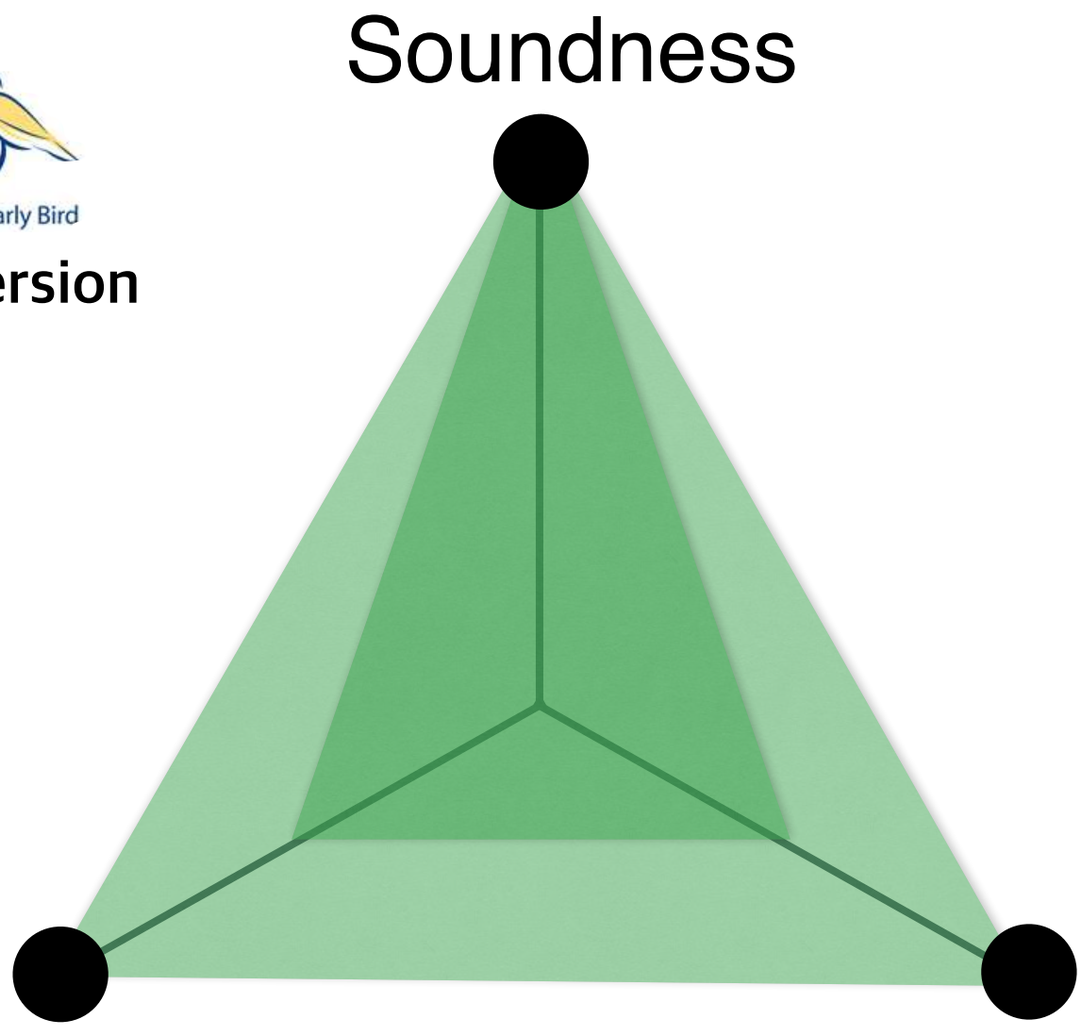


General Sparse Analysis Framework

[TOPLAS'14, PLDI'12]

Our Story: precision

sound-&-global version



General Sparse Analysis Framework

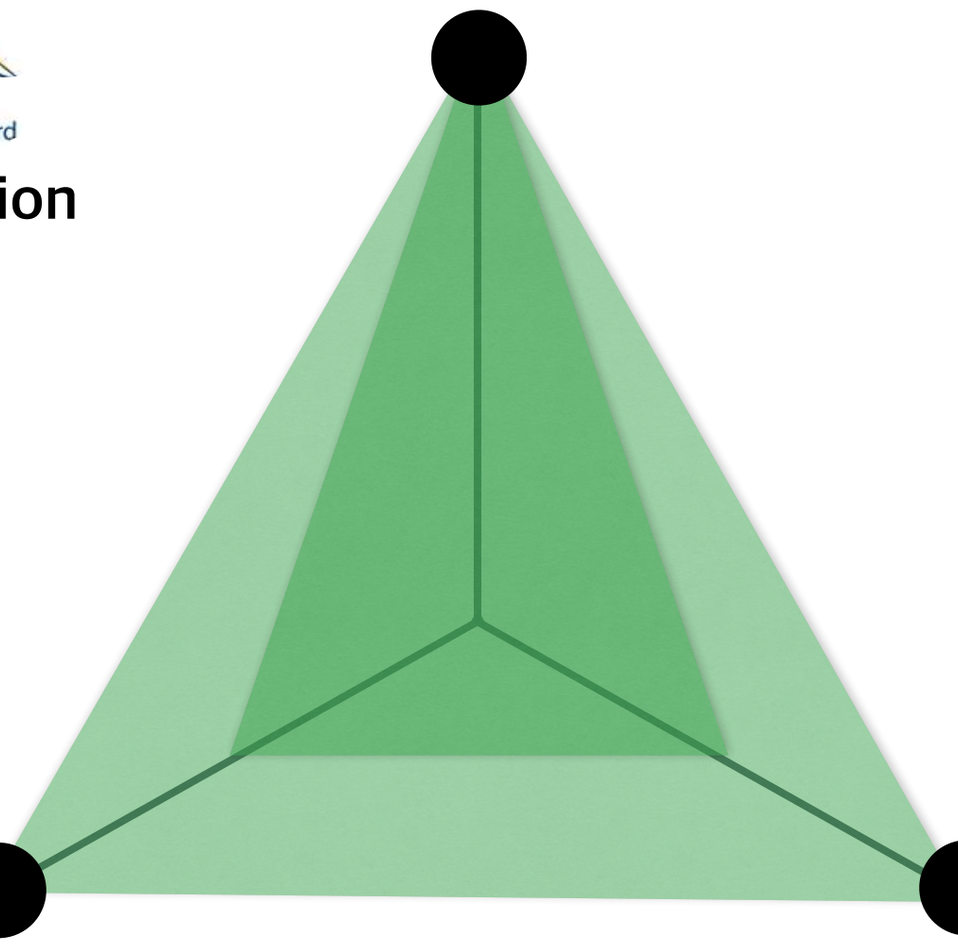
[TOPLAS'14, PLDI'12]

Selective X-Sensitivity Approach

[TOPLAS'15, PLDI'14]

Our Story: verified validator

sound-&-global version



Verified Validator
ZooBerry

General Sparse Analysis Framework
[TOPLAS'14, PLDI'12]

Selective X-Sensitivity Approach
[TOPLAS'15, PLDI'14]

Scalability

- How we achieved
  - sound design of Sparrow
  - spatial & temporal localizations
- Sparse analysis framework
  - general for AI-based analyzers for C-like languages
  - precision-preserving

"An important strength is that the theoretical result is very general. It could be applied to many other analyses. PLDI papers have been accepted that were simply instances of this framework." *(from PLDI reviews)*

- Designed in the *abstract interpretation* framework
- To find memory safety violations in C
  - buffer-overrun, memory leak, null deref., etc.
  - flow-sensitive values analysis for int & ptrs (static + dynamic)
  - for the full set of C

Abstract Semantics

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

$$[\![\hat{P}]\!] \in \mathbb{C} \to \hat{\mathbb{S}} = fix\,\hat{F}$$

$$\mathbb{S} = \mathbb{L} \to \mathbb{V}$$

$$\hat{\mathbb{L}} = Var + AllocSite + AllocSite \times FieldName$$

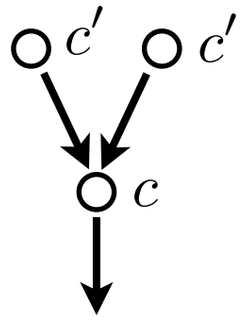
$$\hat{\mathbb{V}} = \hat{\mathbb{Z}} \times 2^{\hat{\mathbb{L}}} \times 2^{AllocSite \times \hat{\mathbb{Z}} \times \hat{\mathbb{Z}}} \times 2^{AllocSite \times 2^{FieldName}}$$

$$\hat{\mathbb{Z}} = \{[l,u] \mid l,u \in \mathbb{Z} \cup \{-\infty, +\infty\} \wedge l \leq u\} \cup \{\bot\}$$

- Abstract semantic function

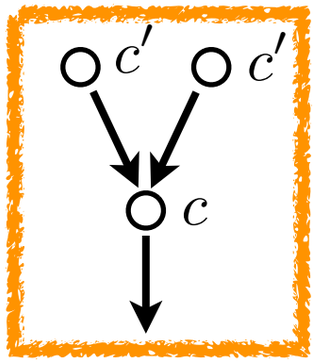
$$\hat{F} \in (\mathbb{C} \to \hat{\mathbb{S}}) \to (\mathbb{C} \to \hat{\mathbb{S}})$$

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



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

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



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

$\hat{f}_c \in \hat{\mathbb{S}} \to \hat{\mathbb{S}}$

$\hat{X} := \hat{X}' := \lambda c.\bot$

repeat

$\quad \hat{X}' := \hat{X}$

$\quad$ **for all** $c \in \mathbb{C}$ **do**

$\quad\quad \hat{X}(c) := \hat{f}_c(\bigsqcup_{c' \hookrightarrow c} X(c'))$

until $\hat{X} \sqsubseteq \hat{X}'$

Naive fixpoint algorithm

$W \in Worklist = 2^{\mathbb{C}}$

$\hat{X} \in \mathbb{C} \to \hat{\mathbb{S}}$

$\hat{f}_c \in \hat{\mathbb{S}} \to \hat{\mathbb{S}}$

$W := \mathbb{C}$

$\hat{X} := \lambda c.\bot$

repeat

$\quad c := \mathsf{choose}(W)$

$\quad \hat{s} := \hat{f}_c(\bigsqcup_{c' \hookrightarrow c} X(c'))$

$\quad$ **if** $\hat{s} \not\sqsubseteq \hat{X}(c)$

$\quad\quad W := W \cup \{c' \in \mathbb{C} \mid c \hookrightarrow c'\}$

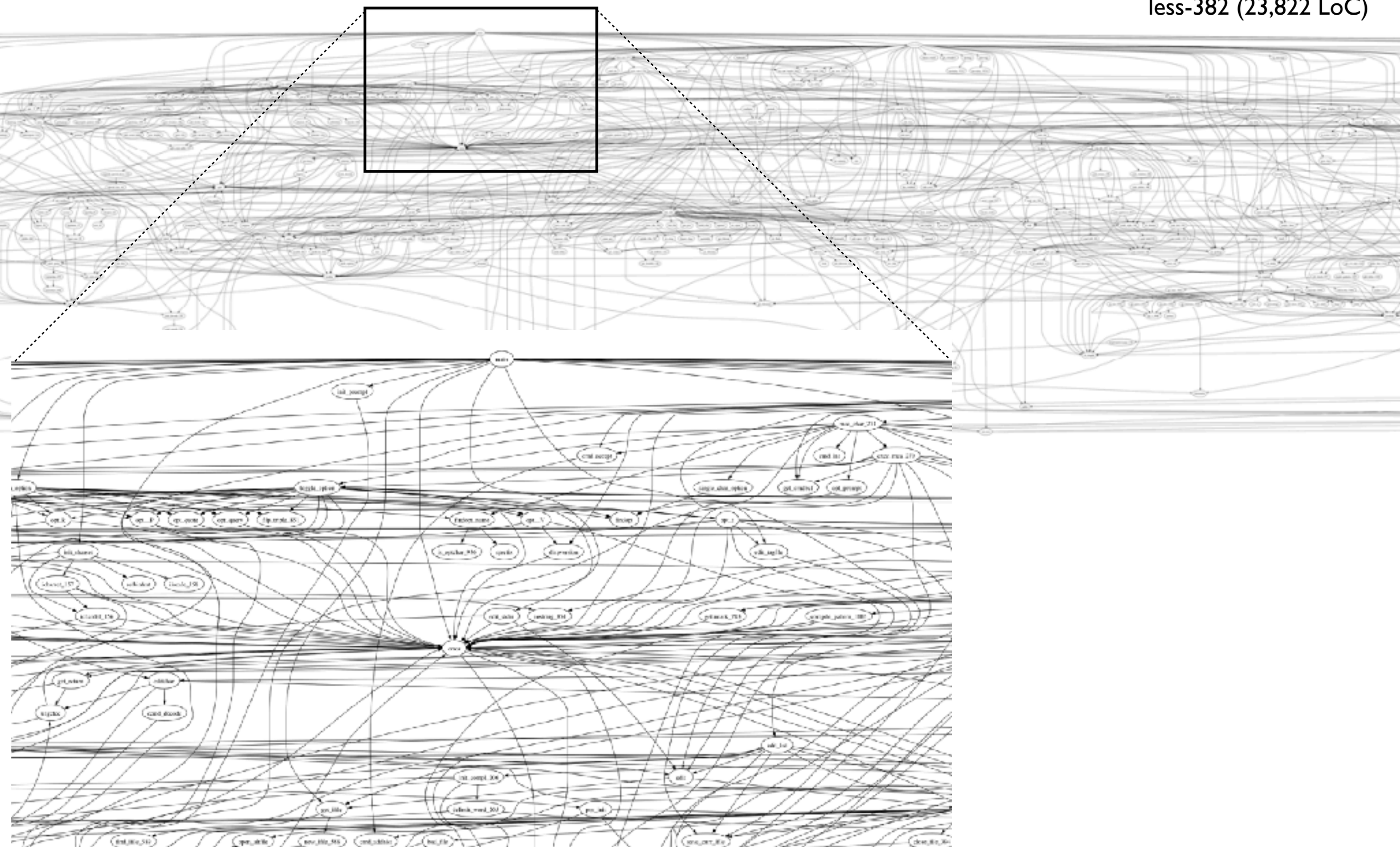
$\quad\quad \hat{X}(c) := \hat{X}(c) \sqcup \hat{s}$

until $W = \emptyset$

Worklist algorithm

The Algorithms Too Weak To Scale

less-382 (23,822 LoC)



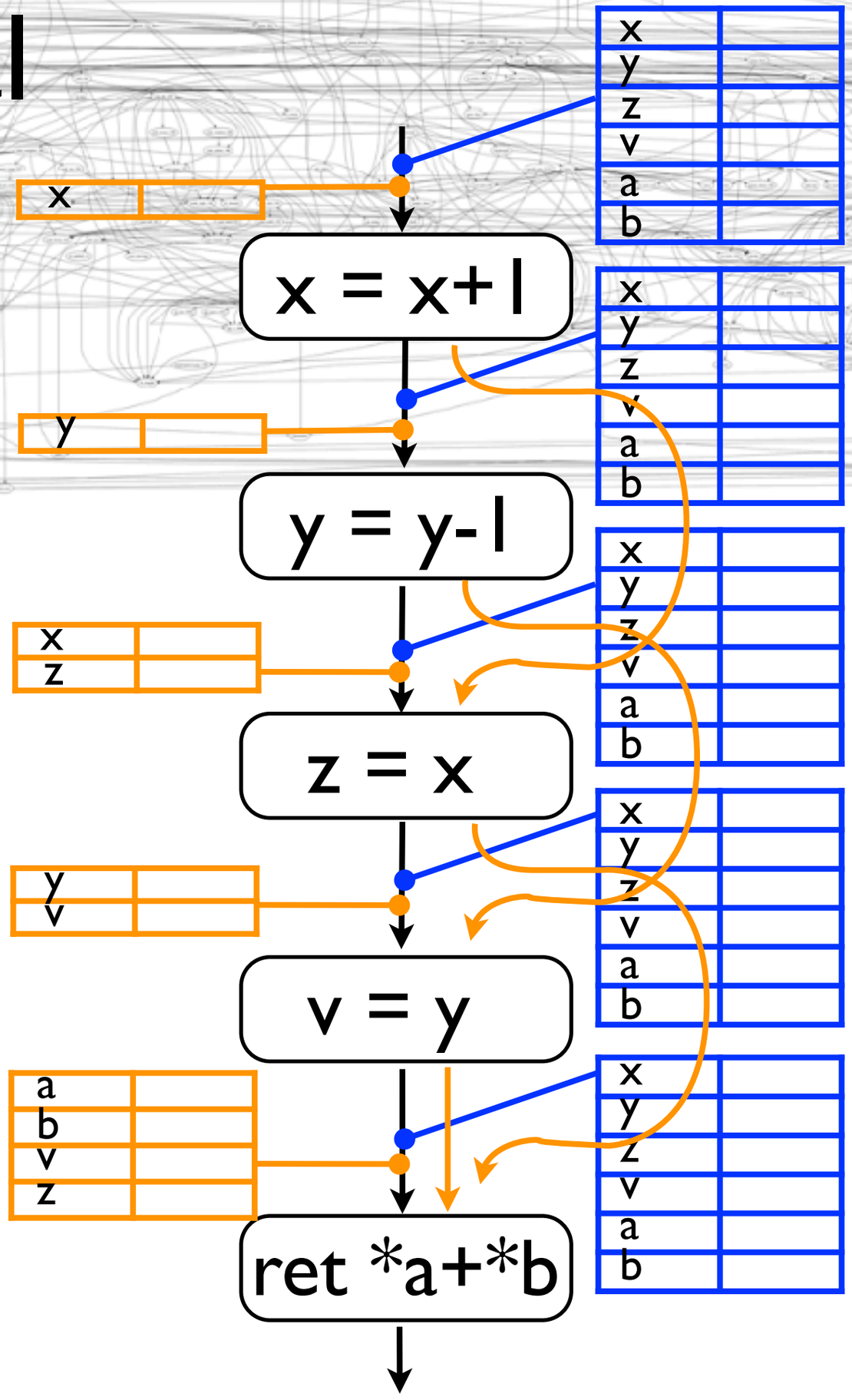
Precision Preserving Sparse Analysis Framework

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

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

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 $\hat{X} \sqsubseteq \hat{X}'$

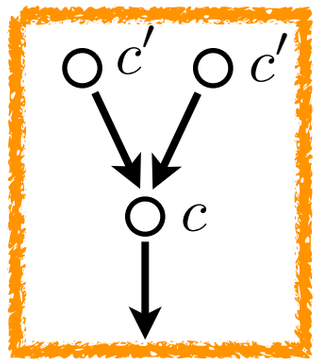


Towards Sparse Version

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

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



- realizable sparse version

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

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

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

Precision Preserving

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

If the following conditions hold

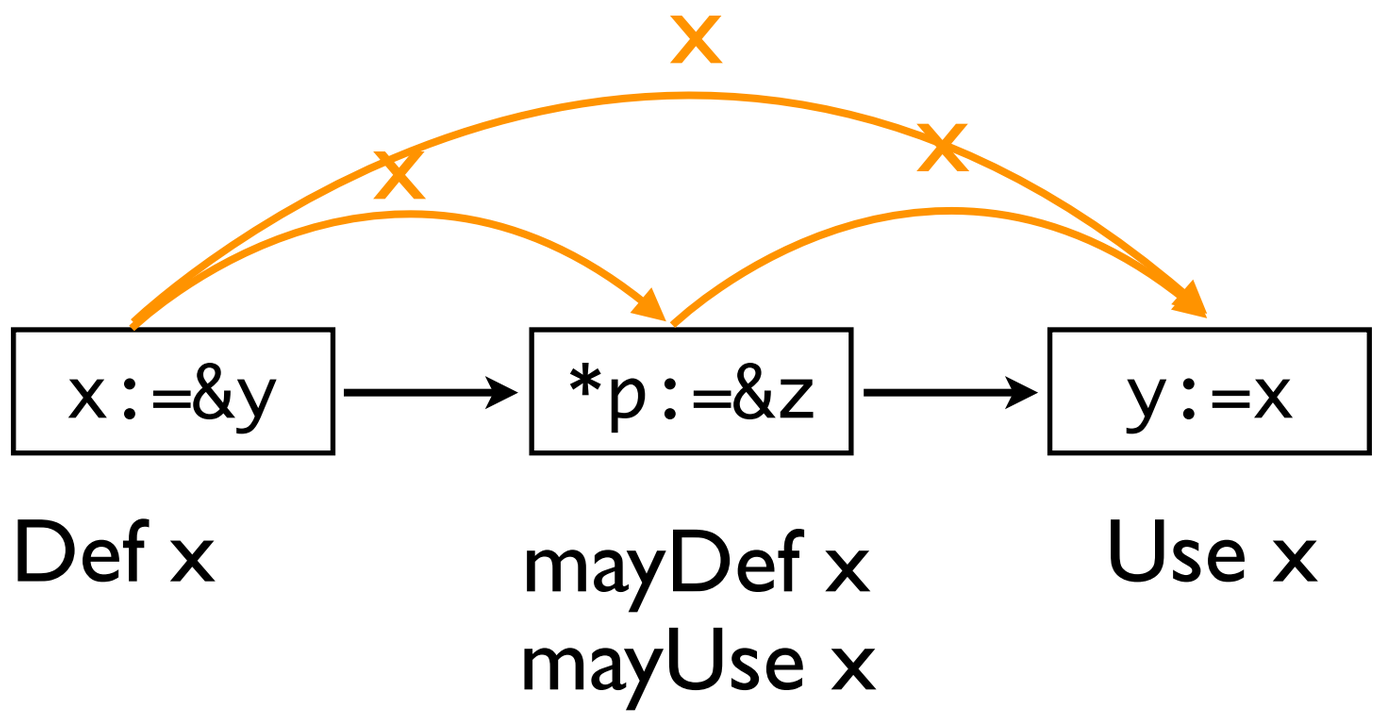
Conditions on $\hat{\mathsf{D}}$ & $\hat{\mathsf{U}}$

- over-approximation

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

- spurious definitions should be also included in uses

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



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

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

Precision Preserving

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

If the previous condition holds.

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

- Yes, by yet another analysis with further abstraction
- correct design

$$\mathbb{C} \to \hat{\mathbb{S}} \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{\mathbb{S}}$$

- abstract semantic function: flow-insensitive

$$\hat{F}_p = \lambda \hat{s}.(\bigsqcup_{c \in \mathbb{C}} \hat{f}_c(\hat{s}))$$

Performance of sound & global

Sparrow
The Early Bird

Programs	LOC	Interval$_{vanilla}$		Interval$_{base}$		Spd↑$_1$	Mem↓$_1$	Interval$_{sparse}$						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

none

spatial localization

spatial+temporal localization

Previous Sparse Techniques

(developed mostly in dfa community)

- Different notion of data dependency

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

  - fail to preserve the original accuracy

- Not general for arbitrary analysis for full C
  - tightly coupled with particular analysis (e.g. pointer analysis for "simple" subsets of C)

ZooBerry System: Automation

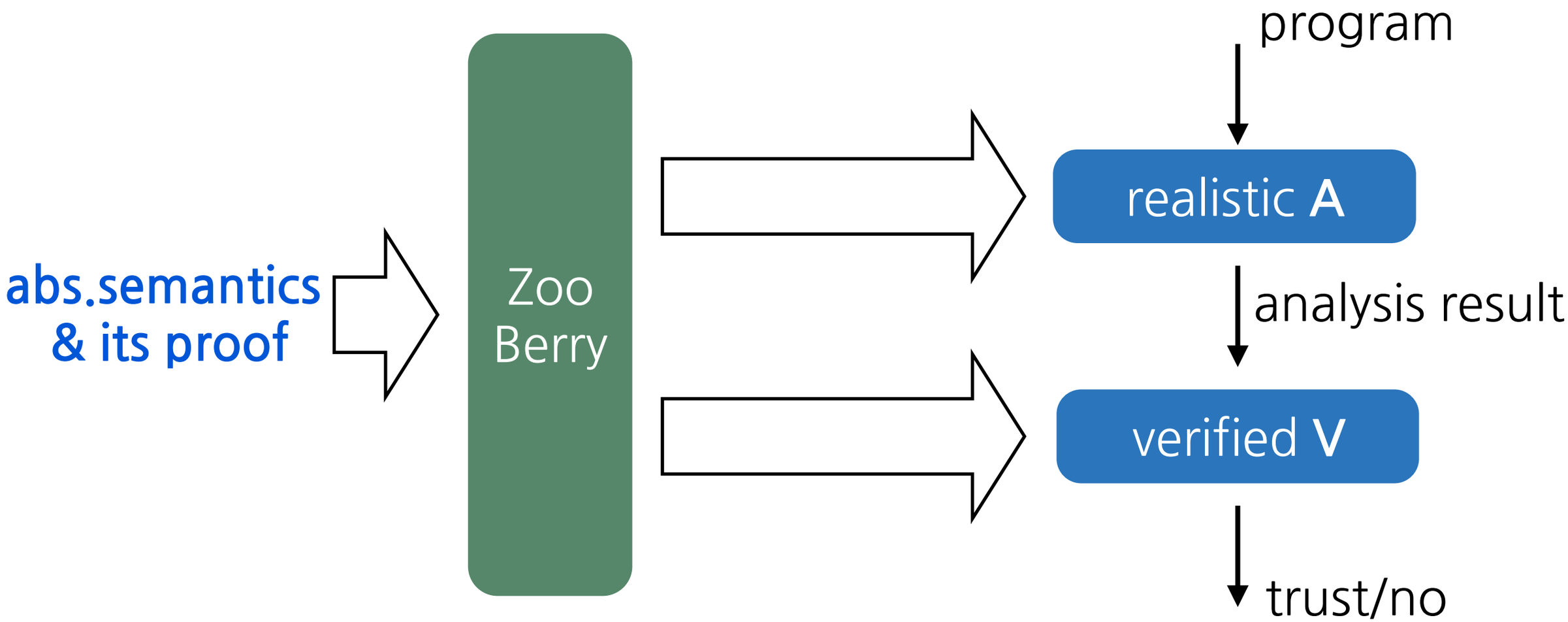
Q: demand grows yet a few hands
A: automation

- general sparse technique
- selective X-sensitive technique
- verified validator

Motivation

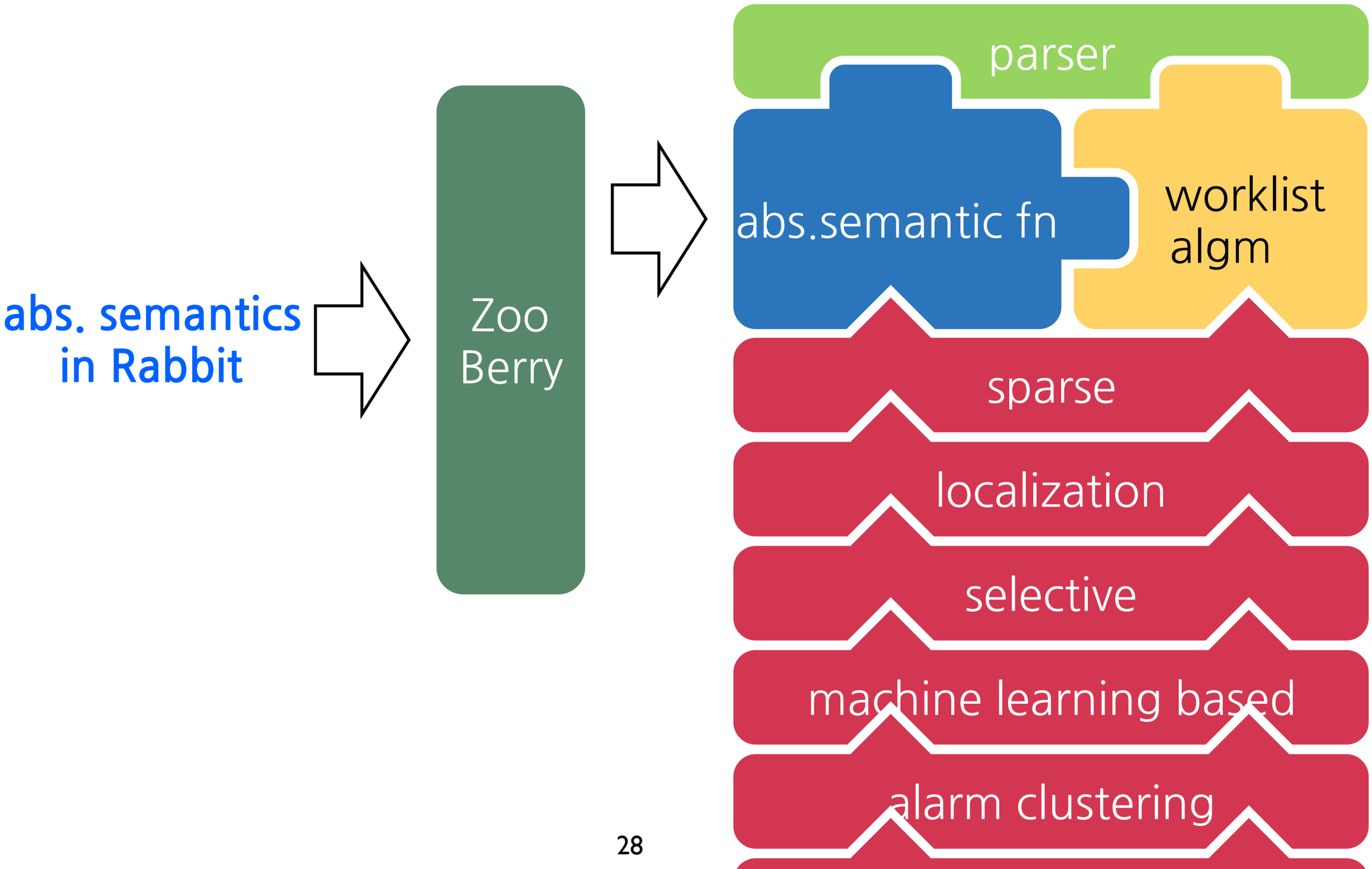
- Let everyone jump-start with
  - high-performance static analyzer **A**
    - their verified validator **V**
- Then with **A**, **V**
  - manually add more eng. to **A** if needed
  - yet keep safe by reusing **V** for validation

Overview

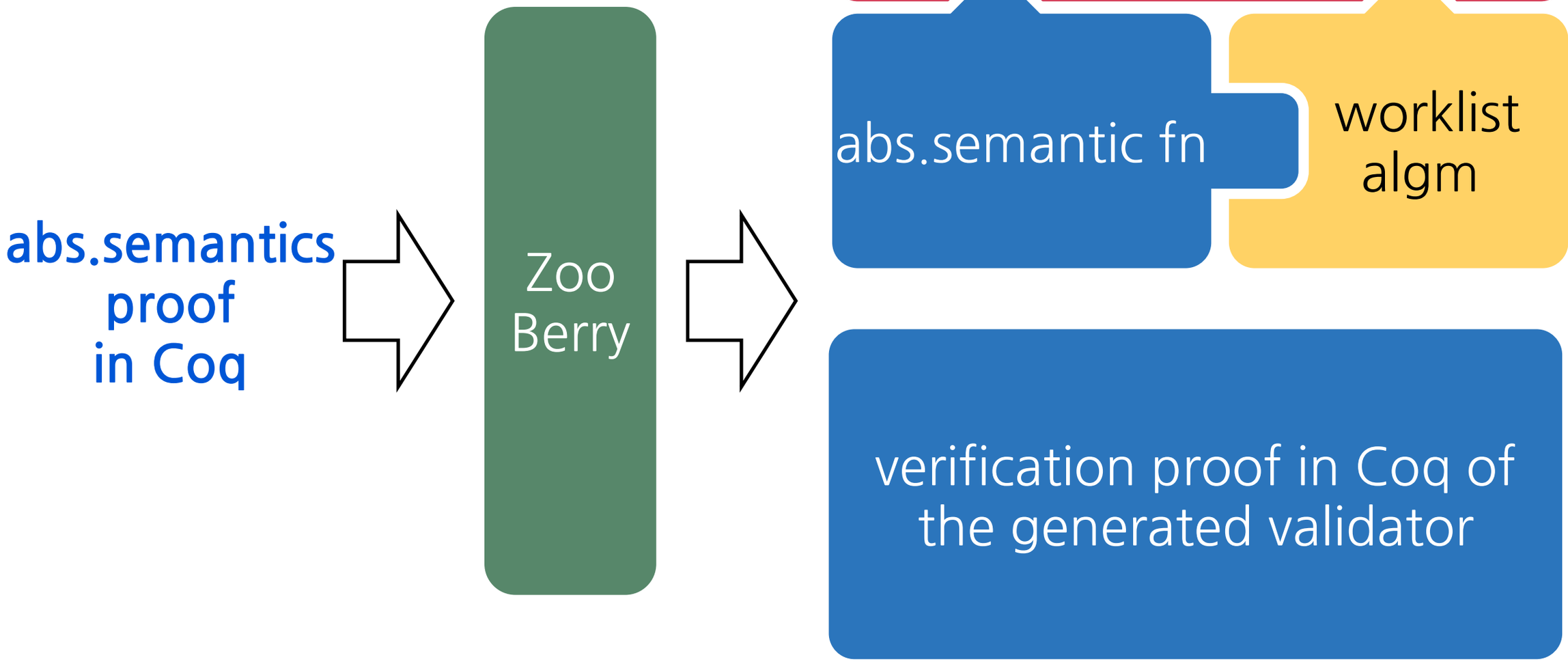


★ = sparsity, selectivity, Zoo, SparrowBerry

Analyzer Generation



Verified Validator Generation



$[\![\cdot]\!]$

$[\![\cdot]\!]^{\wedge}$

$$\frac{}{[\![\cdot]\!] \subseteq [\![\cdot]\!]^{\wedge}}$$

Zoo Berry

program P

realistic **A**

analysis result $\mathcal{X}$

verified **V**

trust/no

$$\frac{}{\forall P, \mathcal{X}.\ \mathbf{V}(P, \mathcal{X}) = \text{yes} \implies [\![\hat{P}]\!] \sqsubseteq \mathcal{X}}$$

Manual vs ZooBerry

Pgm	LOC	manual				ZooBerry				Δ	
		analyzer		validator		analyzer		validator		**analyzr+valdtor**	
		time	mem	time	mem	time	mem	time	mem	**time**	**mem**
time	2K	0	3	0	4	0	4	0	3	NaN	**x1.0**
spell	2K	0	5	0	6	0	5	0	5	NaN	**x0.9**
bc	14K	4	45	10	67	3	50	14	63	**x1.2**	**x1.0**
tar	28K	6	53	21	103	6	86	28	102	**x1.3**	**x1.2**
less	24K	21	144	71	218	23	220	79	323	**x1.1**	**x1.5**
wget	35K	20	118	162	254	31	278	214	306	**x1.3**	**x1.6**
bison	56K	14	120	73	222	19	162	105	208	**x1.4**	**x1.1**
screen	45K	413	780	657	1362	772	2376	705	2224	**x1.4**	**x2.1**
total		478	1268	994	2236	854	3181	1145	3234	**x1.4**	**x1.8**

* unit: sec, MB

Static Analysis in Secrecy

how to analyze programs in cipher-world

Motivation
static-analysis-as-a-service

- help more to enjoy the technology
  - ecology: only two extremes, free vs expensive
  - why not static analysis cloud?

Yet, they are reluctant to upload their source

Fully Homomorphic Encryption (FHE)

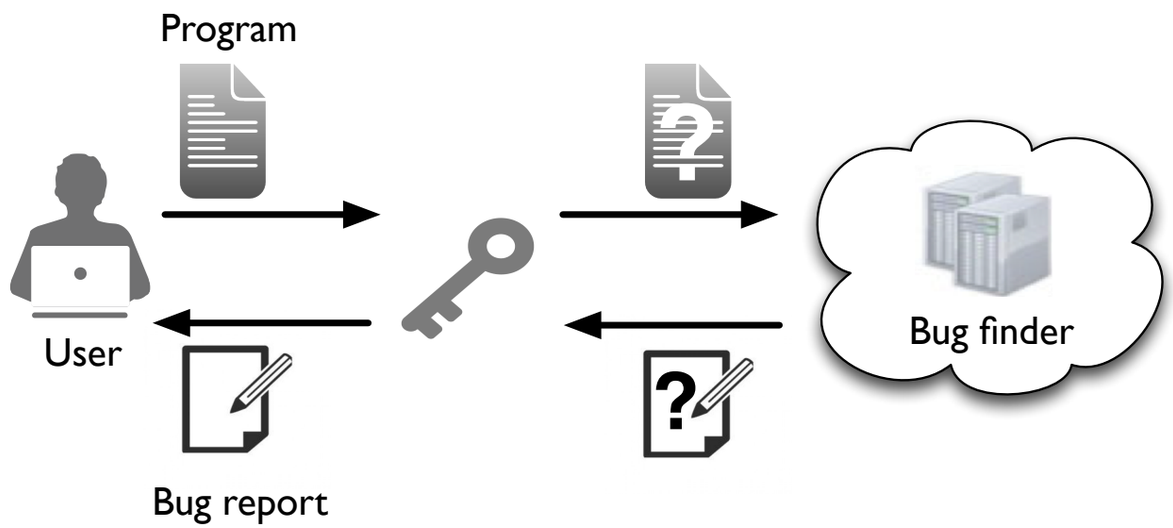
- Encryption $\mathcal{E}$ exists
- For all computable funciton F, exists F' such that

$$F'(\mathcal{E}(m)) = \mathcal{E}(F(m))$$

- For static analyzer A, exists A' such that

$$A'(\mathcal{E}(program)) = \mathcal{E}(A(program))$$

Our Solution: analysis of encrypted program



1. user encrypts programs & send them to cloud
2. cloud analyzes encrypted programs
3. cloud sends encrypted results to the user
4. user decrypts the results

Fully Homomorphic Encryption (FHE)

- A simple example (plaintext $m \in \mathbb{Z}_2 = \{0, 1\}$):

$$\mathcal{E}(m) = m + pq + 2\epsilon$$

random noise for security, much less than p

$$\mathcal{D}(c) = (c \; mod \; p) \; mod \; 2$$

decryption key

- Homorphic

$$\mathcal{E}(m_1) + \mathcal{E}(m_2) \equiv \mathcal{E}(m_1 + m_2)$$

$$\mathcal{E}(m_1) \times \mathcal{E}(m_2) \equiv \mathcal{E}(m_1 \times m_2)$$

Simple Points-to Analysis

- computes a minimal $pt : Var \to 2^{Var}$ satisfying derived constraints.

$$
\begin{array}{rcl}
Program & \to & Stmt^{+} \\
Stmt & \to & x \ := \ y \\
 & | & x \ := \ \&y \\
 & | & x \ := \ *y \\
 & | & *x \ := \ y
\end{array}
$$

$$\frac{x \ := \ y}{pt(x) \supseteq pt(y)} \qquad \frac{x \ := \ \&y}{pt(x) \ni y}$$

$$\frac{x \ := \ *y \qquad pt(y) \ni z}{pt(x) \supseteq pt(z)} \qquad \frac{*x \ := \ y \qquad pt(x) \ni z}{pt(z) \supseteq pt(y)}$$

$$\frac{pt(x) \supseteq pt(y) \qquad pt(y) \ni z}{pt(x) \ni z}$$

Encryption of Programs

- Encrypt by boolean flags
  - 4 kinds of statement constructs
  - each statement involves two vars
  - for every two vars, an encrypted boolean flag

$$
\begin{aligned}
x \ := \ y \quad &\Longleftrightarrow \quad \eta_{yx} = 1 \\
x \ := \ \&y \quad &\Longleftrightarrow \quad \delta_{yx} = 1 \\
x \ := \ *y \quad &\Longleftrightarrow \quad u_{yx} = 1 \\
*x \ := \ y \quad &\Longleftrightarrow \quad v_{xy} = 1
\end{aligned}
$$

Analysis Ops in Encryption

$$\frac{x \;:=\; y}{pt(x) \supseteq pt(y)} \qquad \frac{x \;:=\; \&y}{pt(x) \ni y} \qquad \eta_{yx} = 1 \;\;\&\;\; \delta_{yx} = 1$$

$$\frac{x \;:=\; *y \qquad pt(y) \ni z}{pt(x) \supseteq pt(z)} \qquad \eta_{zx} = \eta_{zx} + \delta_{zy} \times u_{yx}$$

$$\frac{*x \;:=\; y \qquad pt(x) \ni z}{pt(z) \supseteq pt(y)} \qquad \eta_{yz} = \eta_{yz} + \delta_{zx} \times v_{xy}$$

$$\frac{pt(x) \supseteq pt(y) \qquad pt(y) \ni z}{pt(x) \ni z} \qquad \delta_{zx} = \delta_{zx} + \delta_{zy} \times \eta_{yx}$$

Blinded Fixpoint in Encryption

Repeat $m \times m$ times

For $1 \leq j, k \leq m$

$$\eta_{kj} = \eta_{kj} + \sum_{i \neq k,j} \delta_{ki} \times u_{ij} \qquad // \frac{x \texttt{ := } *y \qquad pt(y) \ni z}{pt(x) \supseteq pt(z)}$$

$$\eta_{kj} = \eta_{kj} + \sum_{i \neq k,j} \delta_{ki} \times v_{ij} \qquad // \frac{*x \texttt{ := } y \qquad pt(x) \ni z}{pt(z) \supseteq pt(y)}$$

Repeat $m - 1$ times

For $1 \leq j, k \leq m$

$$\delta_{kj} = \delta_{kj} + \sum_{i \neq k,j} \delta_{ki} \times \eta_{ij} \qquad // \frac{pt(x) \supseteq pt(y) \qquad pt(y) \ni z}{pt(x) \ni z}$$

Experimental Result

- HW : 2.3 GHz Intel i7, 8GB Mem SW : HElib (RLWE-based FHE library)
- Security : 72, Multiplicative depth : 20
- Maximum pointer level 2 in all pgms.
- Not count time for bootstrapping (noise decreasing operation)
- Prototypical, rarely optimized, much room for parallelization.

Program	LOC	# Var	Enc	Propagation	Edge addition	Total	# Bootstrapping
toy	10	9	14s	1h 40m	16m	1h 56m	27
buthead-1.0	46	17	25s	9h 50m	57m	10h 48m	45
cd-discid-1.1	259	41	43s	88h 24m	4h 6m	92h 31m	95

Summing Up

- Practical: Scalable, Sound, Precise, Global Static Analyses
  - General Sparse Analysis Framework for C-like Languages
- Matured: Automation (scalable anlyzr + verified validtr)
  - ZooBerry System
- Possible: Analysis in Secrecy
- Papers: http://kwangkeunyi.snu.ac.kr/publist.html