

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

Hakjoo Oh, Kihong Heo, Wonchan Lee, Woosuk Lee, Kwangkeun Yi
Seoul National University

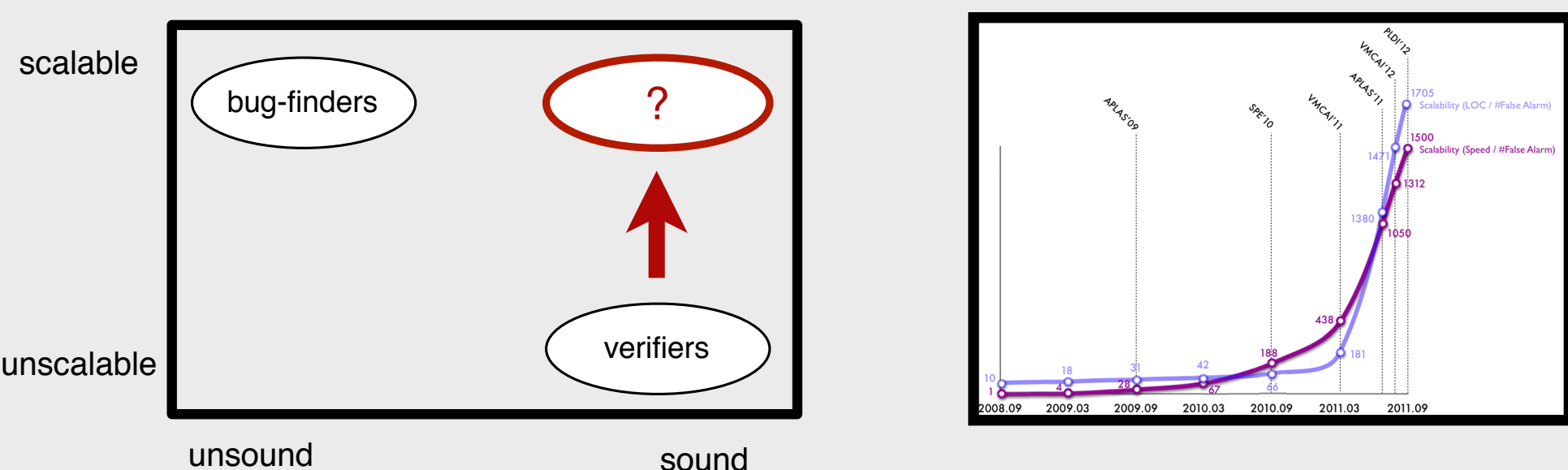
1. Motivation

- Challenge : sound, precise, global yet scalable

-  Sparrow, our commercialized bug-finder

- sound in design
- unsound yet scalable in reality

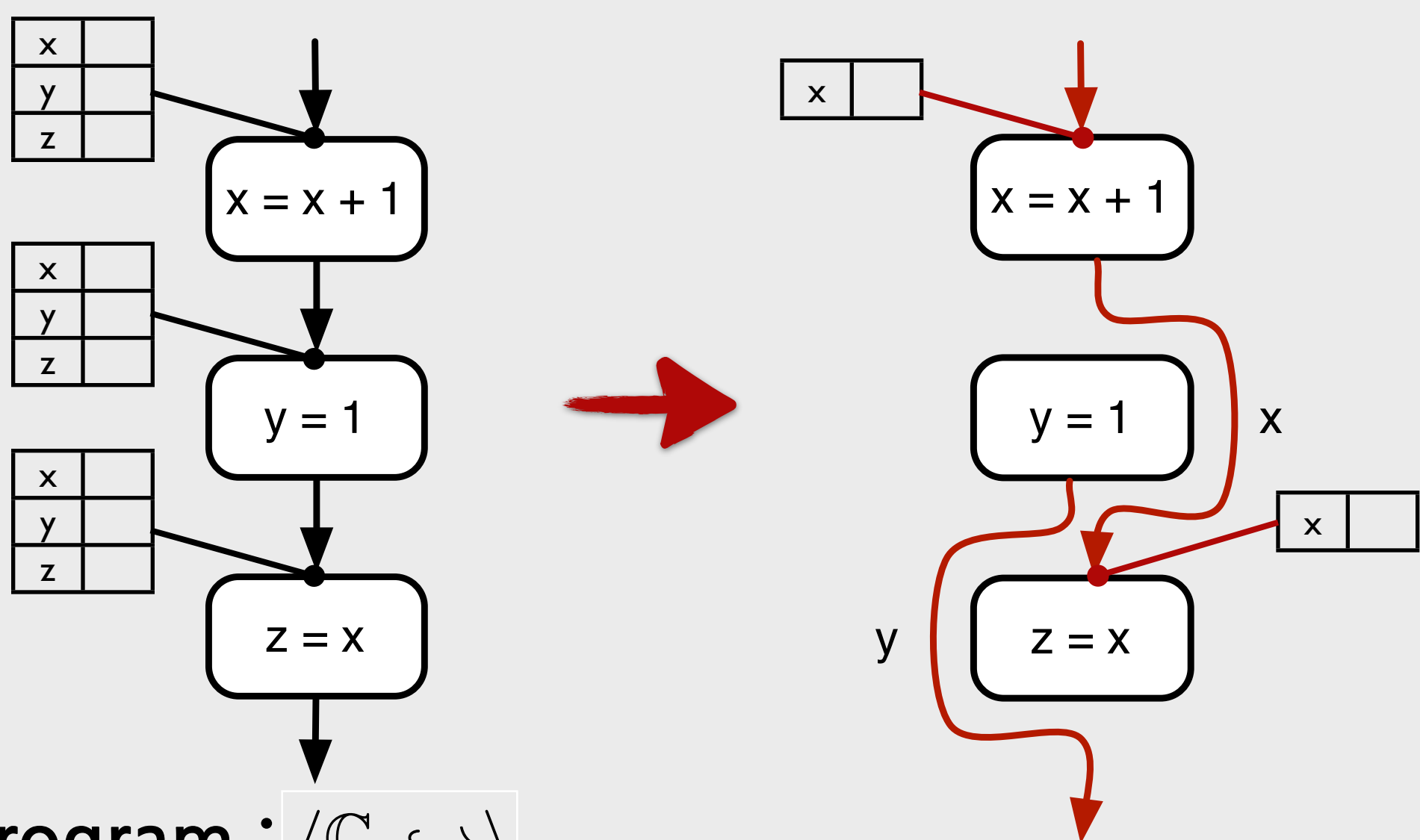
- Goal : scaling up its sound & global version



2. Overview

- Key : **Right Part at Right Moment**

- Follow the data dependency, not the ctrl flow blindly



- Program : $\langle \mathbb{C}, \hookrightarrow \rangle$

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

- Abstract Semantics

$$[\hat{P}] \in \mathbb{C} \rightarrow \hat{S} = \text{fix} \hat{F} \quad \hat{F} \in (\mathbb{C} \rightarrow \hat{S}) \rightarrow (\mathbb{C} \rightarrow \hat{S})$$

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

- Precision Preserving Framework

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

$$\text{fix} F = \text{fix} F_s$$

5. Conclusions

- A **general framework** for sparse analysis
 - data dependency
 - safe & correct def/use sets

3. Sparse Analysis Framework

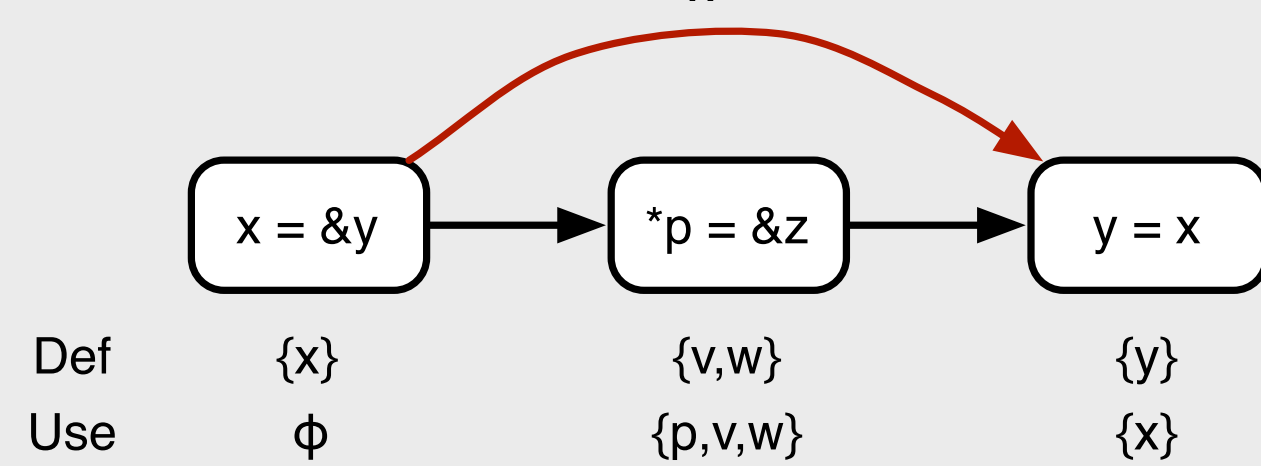
- Data Dependency

$$c_0 \xrightarrow{l} c_n \triangleq \exists c_0 \dots c_n \in \text{Path}, l \in \mathbb{L}.$$

$$l \in D(c_0) \cap U(c_n) \wedge \forall i \in (0, n). l \notin D(c_i)$$

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

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



- Approximated Data Dependency

$$c_0 \xrightarrow{l} c_n \triangleq \exists c_0 \dots c_n \in \text{Path}, l \in \mathbb{L}.$$

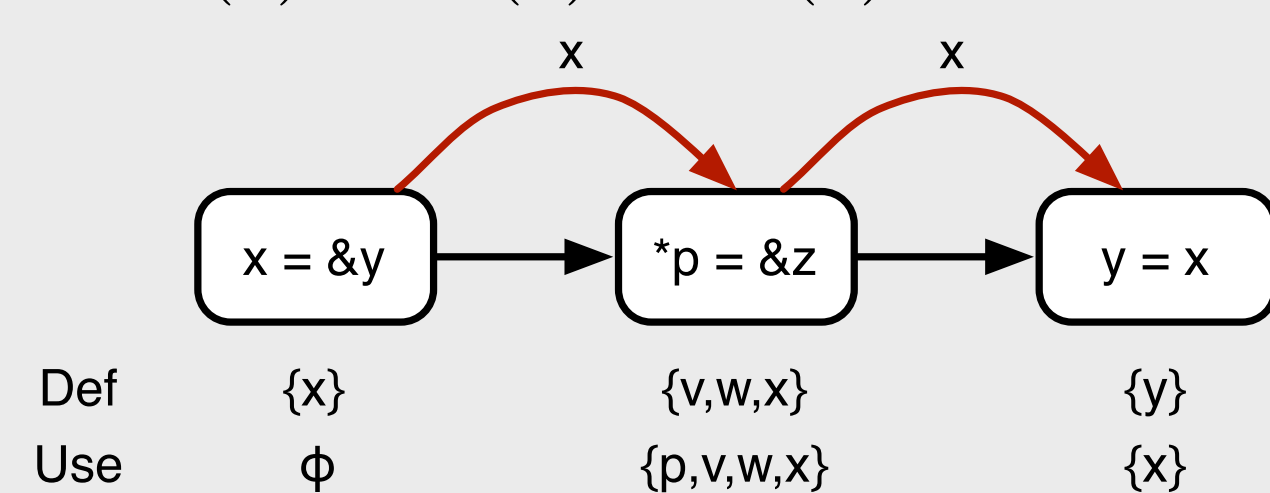
$$l \in \hat{D}(c_0) \cap \hat{U}(c_n) \wedge \forall i \in (0, n). l \notin D(c_i)$$

- over-approximation

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

- spurious definitions should be included in uses

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



4. Experiments

- Interval analysis

Program	LOC	Vanilla		Sparse		Spd \uparrow	Mem \downarrow
		Time	Mem	Time	Mem		
make-3.76.1	27K	24240	1391	21	114	1154x	92%
wget-1.9	35K	44092	2546	11	85	4008x	97%
screen-4.0.2	45K	∞	∞	767	303	N/A	N/A
emacs-22.1	400K	∞	∞	37,830	7795	N/A	N/A
linux-3.0	710K	∞	∞	33,618	2,0529	N/A	N/A
ghostscript-0.00	1,363K	∞	∞	14,814	6384	N/A	N/A

- Octagon analysis

Program	LOC	Vanilla		Sparse		Spd \uparrow	Mem \downarrow
		Time	Mem	Time	Mem		
gzip-1.2.4a	7K	2078	2832	21	269	98x	91%
bc-1.06	13K	9536	6987	55	358	173x	95%
make-3.76.1	28K	∞	∞	331	666	N/A	N/A
wget-1.9	35K	∞	∞	288	646	N/A	N/A
screen-4.02	45K	∞	∞	1,6433	9199	N/A	N/A
sendmail-8.13.6	130K	∞	∞	6,4808	2,9658	N/A	N/A