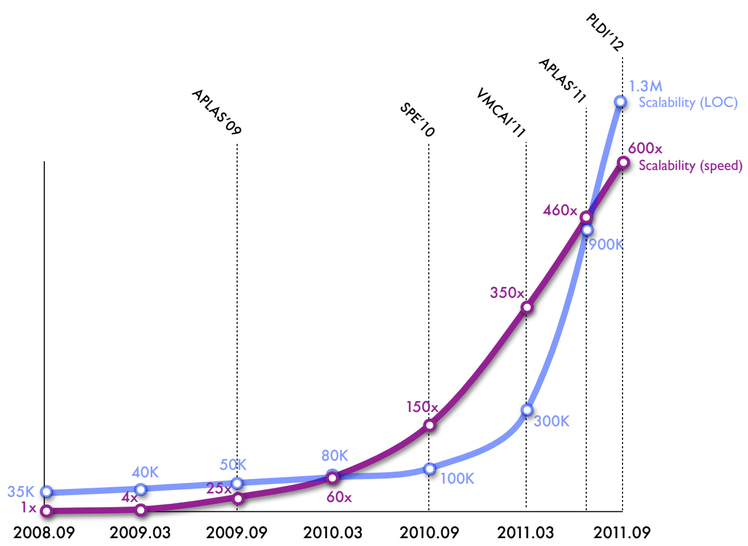
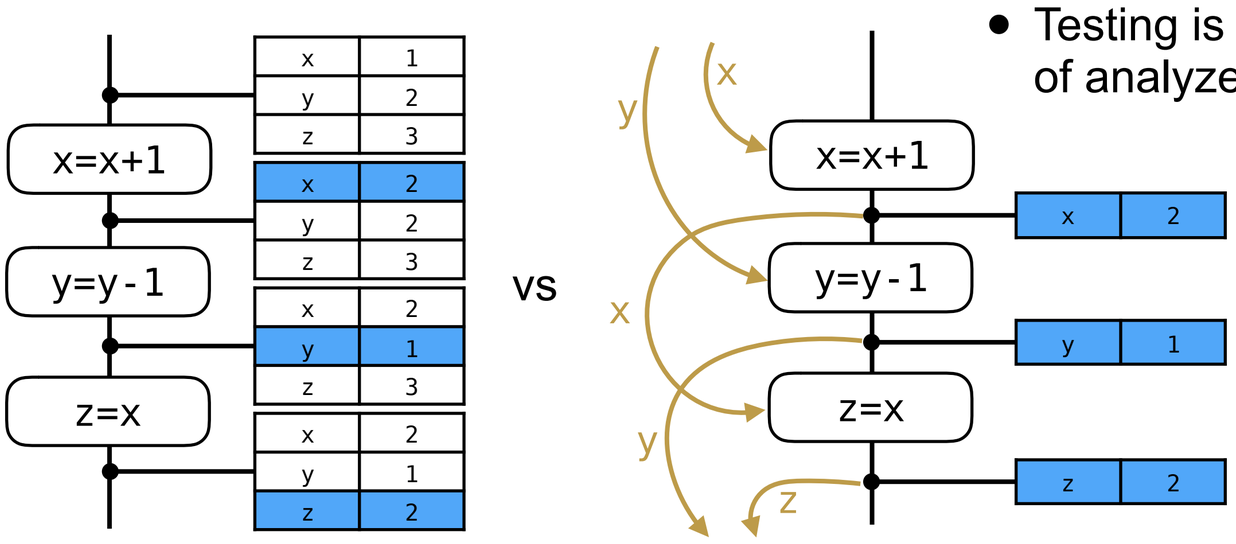
ZooBerry: Automatic Generation of Sparse Global Static Analyzers & Their Validators

Sungkeun Cho and Kwangkeun Yi
Seoul National University

Background



Key: sparse analysis[1]



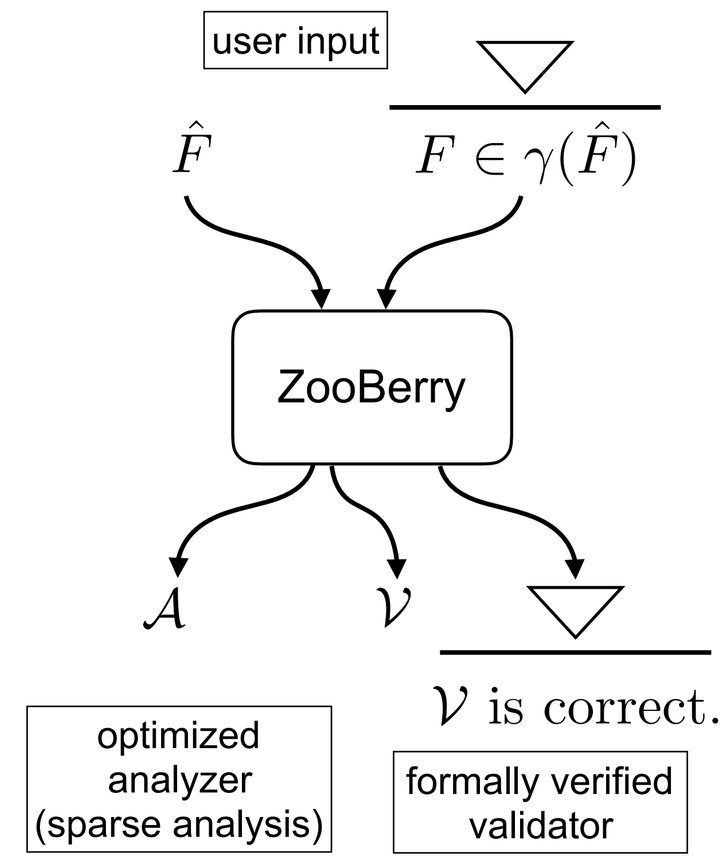
Problem

Applying the sparse analysis is,

1. burdensome
   - pre-analysis for drawing data dependency graph
   - additional 1500 LOC in OCaml (SparseSparrow[1]: our buffer overrun analyzer for C)
2. (prone to be) buggy
   - 5 of 13 bugs we found[2] from SparseSparrow were about the sparse analysis.
   - Testing is not good to find bugs of analyzers.

Solution

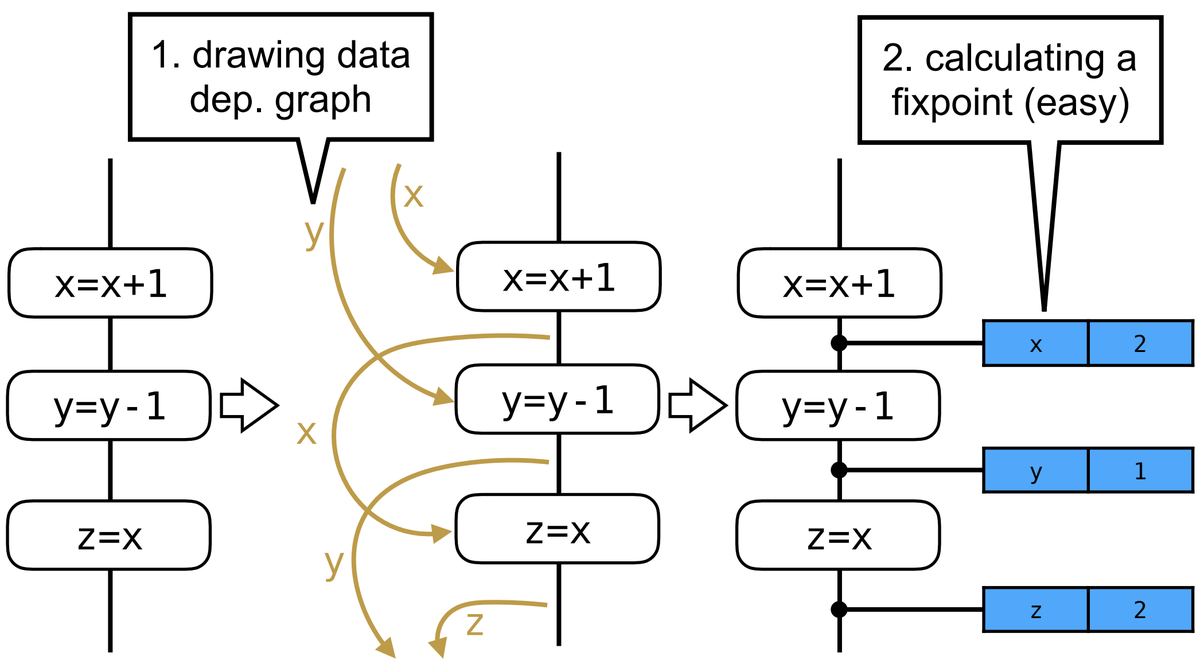
Automatic Generation!



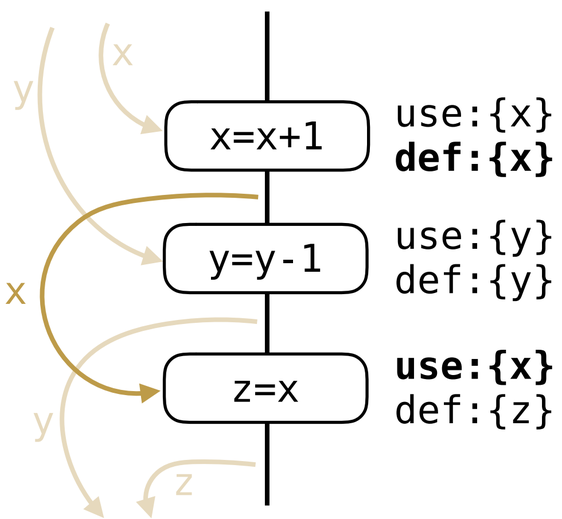
Automatically implementing the verified validation technique

Automatically implementing the sparse analysis technique

Applying the sparse analysis



access information
set of locations that are defined/used during analyses



For sparsity, we designed a type-based transformation of the analysis spec.

$$\hat{f} : \mathrm{Cmd} \to \hat{\mathbb{M}} \to \hat{\mathbb{M}}$$

$$\hat{f}' : \mathrm{Cmd} \to \hat{\mathbb{M}} \to (\hat{\mathbb{M}} \times \mathrm{Acc})$$

$$\mathrm{Acc} = 2^{\hat{\mathbb{L}}} \times 2^{\hat{\mathbb{L}}}$$

Correctness (naive)

A validation is simply a fixpoint check.

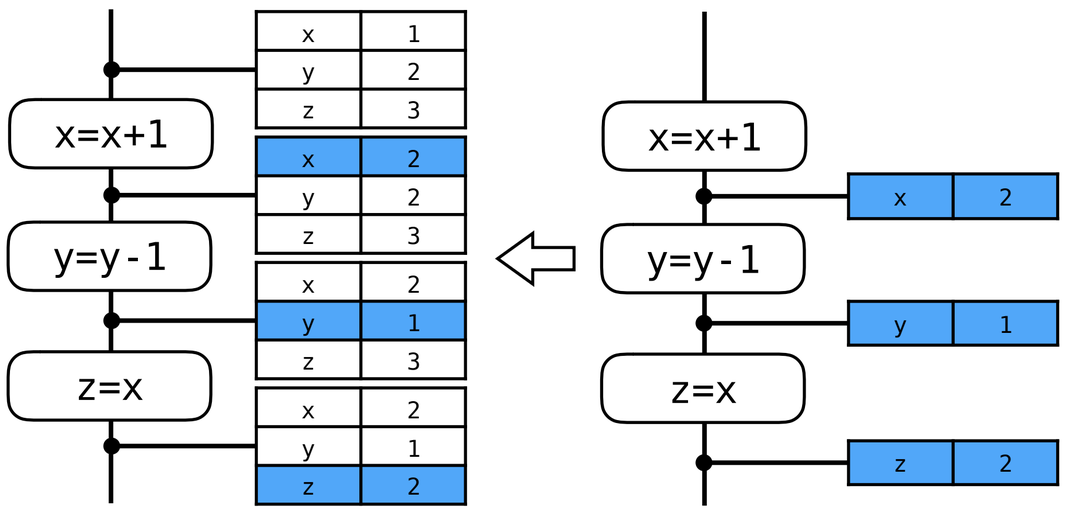
$$\forall p, \hat{s} : \underline{\mathcal{V}(p, \hat{s}) = \texttt{success}} \Rightarrow [\![p]\!] \in \gamma(\hat{s})$$

$$\hat{F}(\hat{s}) \sqsubseteq \hat{s}$$

However, $\hat{f}$ (y=y-1 , x 2) $\not\sqsubseteq$ y 1

Densification

"Recover the original analysis result and validate it."



1. We designed an efficient densification algorithm.
2. The validator checks also the densification is correct.

Correctness (revised)

$$\forall p, \hat{s}, \mathcal{D}(\hat{s}) : \mathcal{V}(p, \hat{s}, \mathcal{D}(\hat{s})) = \texttt{success} \;\Rightarrow\; [\![p]\!] \in \gamma(\mathcal{D}(\hat{s})) \;\wedge\; \hat{s} \subseteq^{\mathrm{use}}_{p} \mathcal{D}(\hat{s})$$

References

[1] Design and Implementation of Sparse Global Analyses for C-like Lnaguages. Hakjoo Oh, Kihong Heo, Wonchan Lee, Woosuk Lee, and Kwangkeun Yi. *PLDI'12*.
[2] Towards Scalable Translation Validation of Static Analyzers. Jeehoon Kang, Sungkeun Cho, Joonwon Choi, Chung-Kil Hur, and Kwangkeun Yi. *ROSAEC techmemo 2014.* http://rosaec.snu.ac.kr/publish/2014/techmemo/ROSAEC-2014-003.pdf