Minimally Comparing Relational Abstract Domains

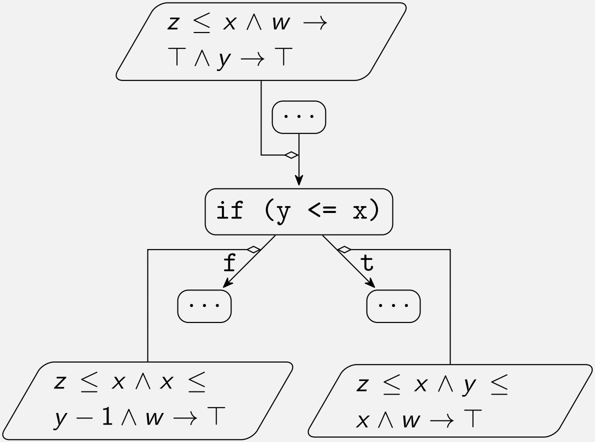
Kenny Ballou
Elena Sherman

Boise State University
Boise, Idaho
United States of America

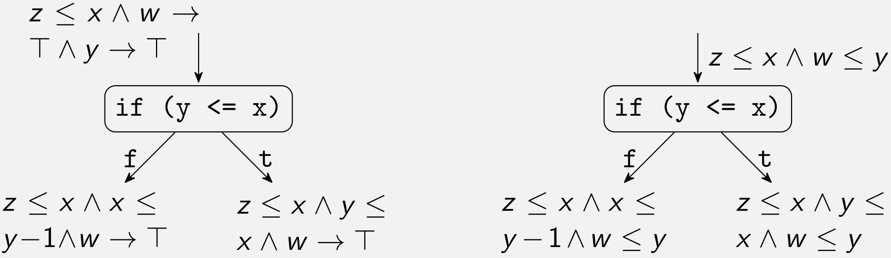
October 2023

Outline

Static analysis computes information about programs



Researchers compare invariants to select efficient and precise analyses



(a) Original Analysis (b) Improved Analysis

Researchers compare invariants to select efficient and precise analyses

$z \to [0, +\infty) \wedge$
$x \to [0, +\infty) \wedge$
$y \to (-\infty, +\infty) \wedge$
$w \to (-\infty, +\infty)$

```
if (y <= x)
```

f:

$z \to [0, +\infty) \wedge$
$x \to [0, +\infty) \wedge$
$y \to [1, +\infty) \wedge$
$w \to (-\infty, +\infty)$

t:

$z \to [0, +\infty) \wedge$
$x \to [0, +\infty) \wedge$
$y \to (-\infty, +\infty) \wedge$
$w \to (-\infty, +\infty)$

(a) Example Interval Analysis

$z \to \{+\} \wedge x \to \{0, +\} \wedge y \to \top \wedge w \to \top$

```
if (y <= x)
```

f:

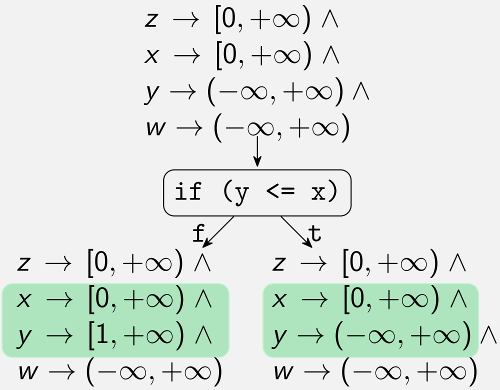
$z \to \{+\} \wedge$
$x \to \{0, +\} \wedge$
$y \to \{+\} \wedge$
$w \to \top$

t:

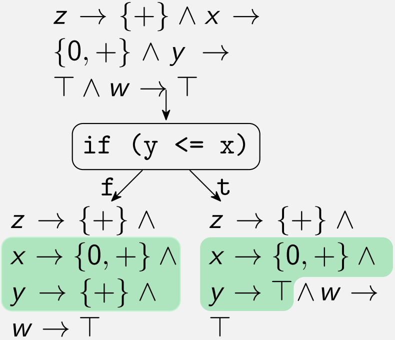
$z \to \{+\} \wedge$
$x \to \{0, +\} \wedge$
$y \to \top \wedge w \to \top$

(b) Example Predicate Analysis

Researchers compare invariants to select efficient and precise analyses

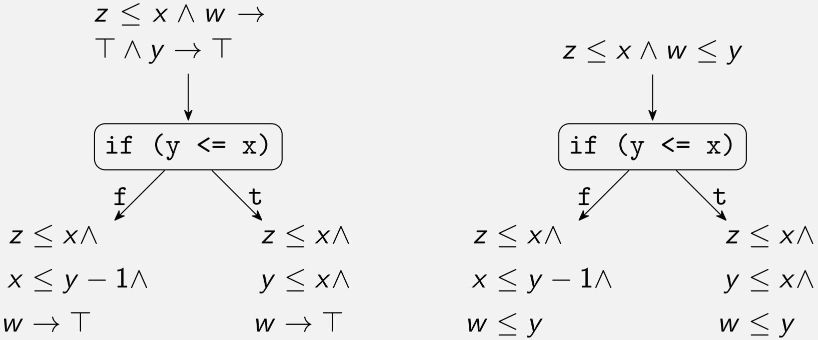


(a) Example Interval Analysis



(b) Example Predicate Analysis

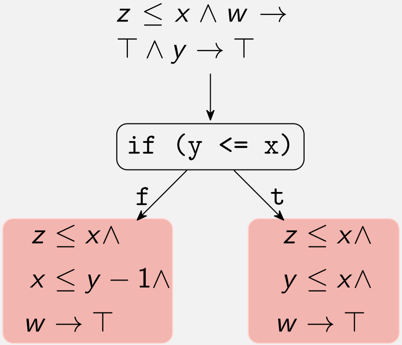
Comparing relational states is non-trivial



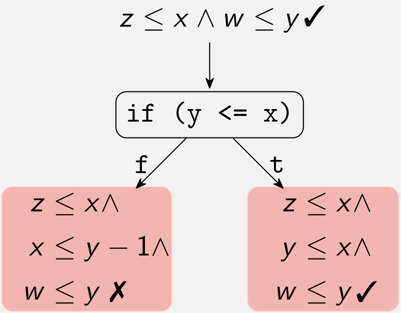
(a) Original Relational Analysis

(b) Improved Relational Analysis

Comparing relational states is non-trivial

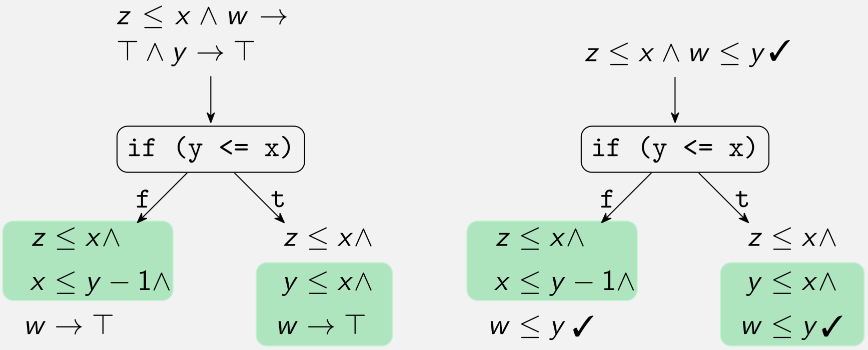


(a) Original Relational Analysis



(b) Improved Relational Analysis

Comparing relational states is non-trivial



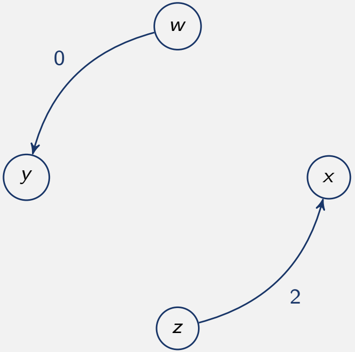
(a) Original Relational Analysis

(b) Improved Relational Analysis

Abstract Domains

Zone Abstract Domain

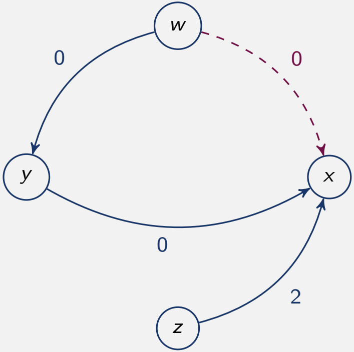
$$z - x \leq 0$$
$$w - y \leq 0$$



Abstract Domains

Zone Abstract Domain

$$\begin{array}{r} z - x \leq 0 \\ w - y \leq 0 \\ y - x \leq 0 \\ \hline w - x \leq 0 \end{array}$$

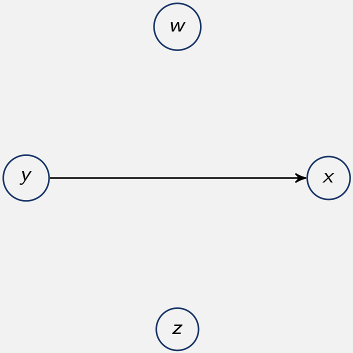


Abstract Domains

Symbolic Predicates [1]

$$\frac{\begin{array}{r} z \to \{+\} \\ x \to \{0, +\} \\ y \to \{-, 0, +\} \\ w \to \{-, 0, +\} \end{array}}{y \leq x}$$

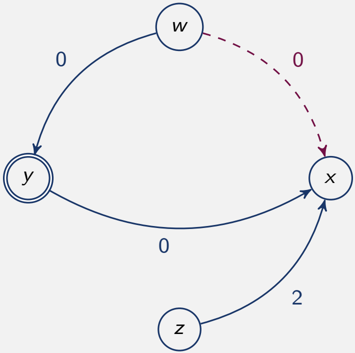
(a) Symbolic Predicates (Sign Domain)



(b) Relational projection of Symbolic Predicates

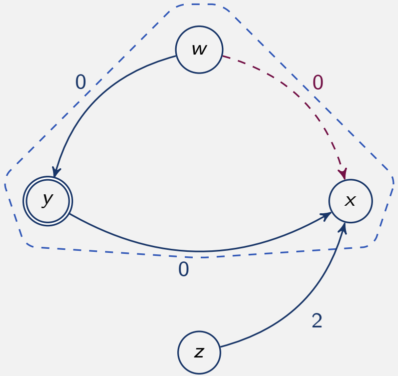
[1] Sherman and Dwyer, "Exploiting Domain and Program Structure to Synthesize Efficient and Precise Data Flow Analyses (T)"

Identifying minimal changes within Zone states [2]



[2] Ballou and Sherman, "Identifying Minimal Changes in the Zone Abstract Domain"

Identifying minimal changes within Zone states [2]



[2] Ballou and Sherman, "Identifying Minimal Changes in the Zone Abstract Domain"

Formula are compared via logical entailment

Forward

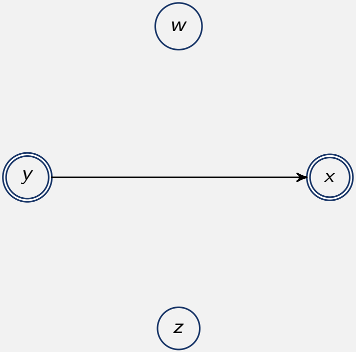
```
(push)
(forall ((w Int) (x Int) (y Int) (z Int))
        (assert (=> (and (<= z y) (<= y x))
                    (and (<= z y) (<= y x) (<= w x)))))
(check-sat)
(pop)
```

Backward

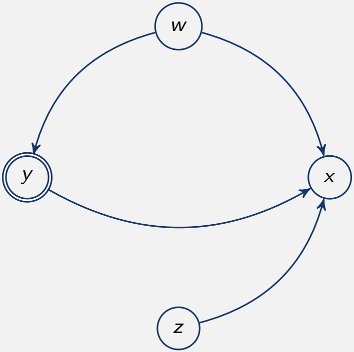
```
(push)
(forall ((w Int) (x Int) (y Int) (z Int))
        (assert (=> (and (<= z y) (<= y x) (<= w x))
                    (and (<= z y) (<= y x)))))
(check-sat)
(pop)
```

Minimal union between two sets of invariants

Example



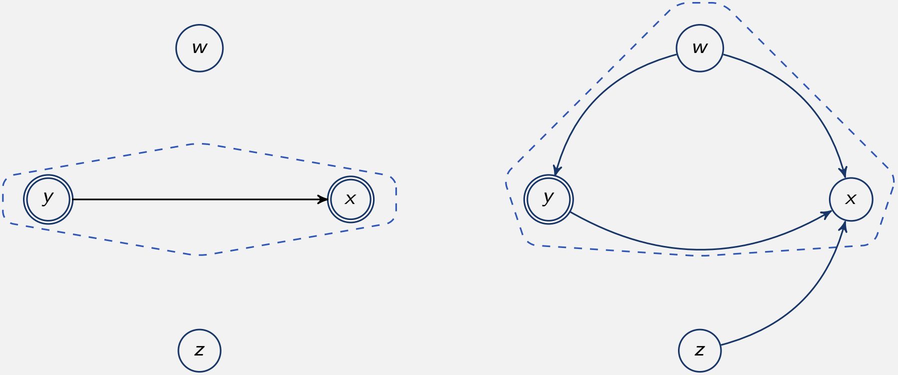
(a) Symbolic Predicate State



(b) Zone State

Minimal union between two sets of invariants

Example



(a) Symbolic Predicate State

(b) Zone State

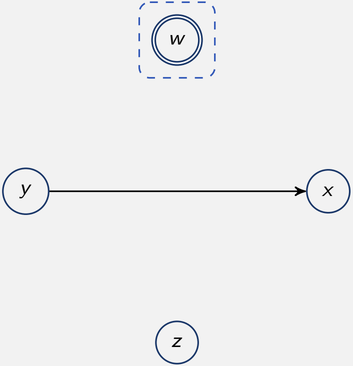
$S_1 = \{x, y\}$

$S_1 \subset S_2$

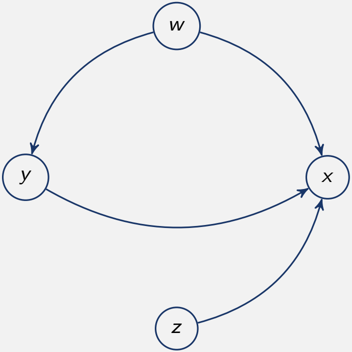
$S_2 = \{w, x, y\}$

Minimal union between two sets of invariants

Example



(a) Symbolic Predicate State



(b) Zone State

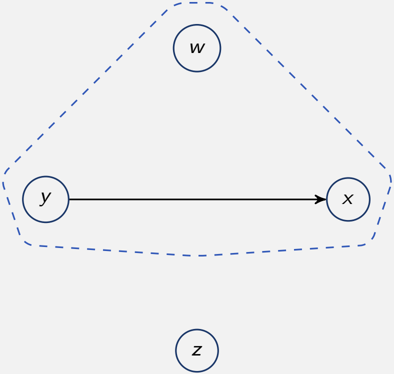
$S_1 = \{w\} \cup S_1'$

$S_1 \stackrel{?}{=} S_2$

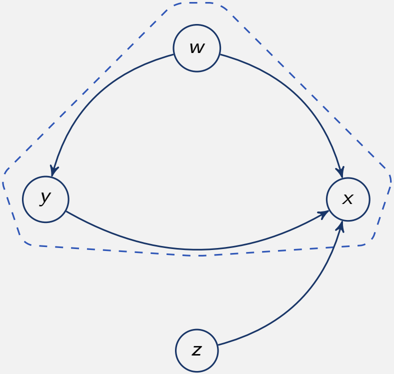
$S_2 = \{w, x, y\}$

Minimal union between two sets of invariants

Example



(a) Symbolic Predicate State



(b) Zone State

$S_1 = \{w, x, y\}$

$S_1 \equiv S_2$

$S_2 = \{w, x, y\}$

Experimental Evaluation

Research Questions

- RQ1 Does our technique affect the invariant comparison between different analysis techniques for the same abstract domain?
- RQ2 Does our technique affect the invariant comparison between two different relational abstract domains?
- RQ3 How effective and efficient is our algorithm on real-world invariant comparisons?

Experimental Evaluation

Research Questions

- RQ1 Does our technique affect the invariant comparison between different analysis techniques for the same abstract domain?
- RQ2 Does our technique affect the invariant comparison between two different relational abstract domains?
- RQ3 How effective and efficient is our algorithm on real-world invariant comparisons?

Subject Programs

192 Java methods selected from previous research

Experimental Evaluation

Research Questions

RQ1 Does our technique affect the invariant comparison between different analysis techniques for the same abstract domain?

RQ2 Does our technique affect the invariant comparison between two different relational abstract domains?

RQ3 How effective and efficient is our algorithm on real-world invariant comparisons?

Subject Programs

192 Java methods selected from previous research

Experiments

- Compared Zones with different parameters around widening
- Compared Zones vs. Symbolic Predicates

Comparing widening parameters on Zones

Zones, widening after 2 iterations vs. widening after 5 iterations

Comparison	$Z \equiv Z_{k=5}$	$Z \prec Z_{k=5}$
Full	6555	9
Minimal	6562	2

Zones widening after 2 iterations vs. threshold widening after 2 iterations

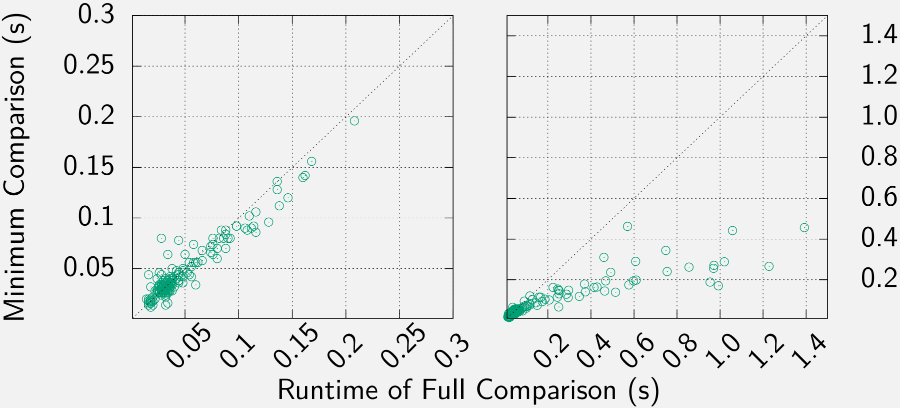
Comparison	$Z \equiv Z_{ths}$	$Z \prec Z_{ths}$
Full	6519	45
Minimal	6545	19

Comparing Zones to Symbolic Predicates

Zones with threshold widening vs. Symbolic Predicates

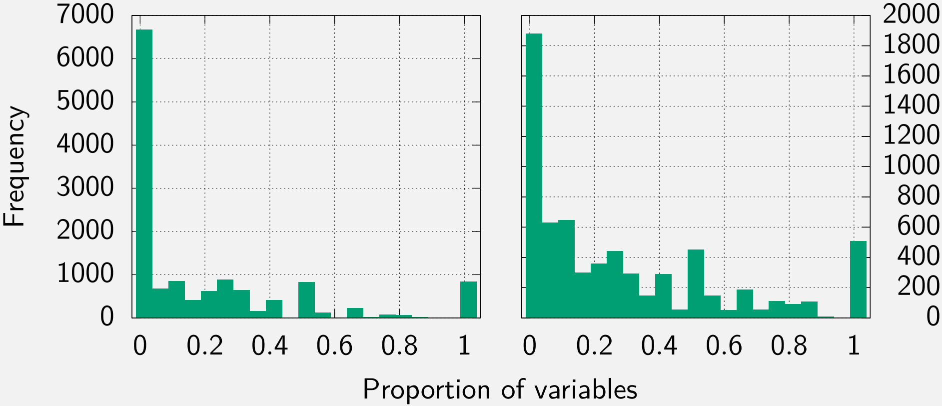
Comparison	$Z_{ths} \equiv P$	$Z_{ths} \prec P$	$Z_{ths} \succ P$	$Z_{ths} \prec\succ P$	$Z_{ths}\, ?\, P$
Full	1227	3173	196	1947	21
Minimal	3675	2353	248	288	0

Walltime of comparisons between full vs. minimal



(left): Zones with standard widening compared to zones with threshold widening, (right): Zones with threshold widening vs. symbolic predicates.

Comparison of variable reductions per comparison



(left): Zones with standard widening compared to zones with threshold widening, (right): Zones with threshold widening vs. symbolic predicates.

Conclusion

Experimental Results

- Demonstrated a minimal union algorithm for comparing relational abstract domains, eliminating *carry-over* effects
- Enables more precise comparison between techniques and relational abstract domains
- Empirical evaluations show the algorithm is effective and efficient

Future Work

- Extend to other Weakly-Relational Domains, e.g., Octagons
- Optimize union to consider the pre-order relation between domains

Thank you

Questions?

The work reported here was supported by the U.S. National Science Foundation under award CCF-19-42044.

References I

[1] Kenny Ballou and Elena Sherman. "Identifying Minimal Changes in the Zone Abstract Domain". In: *Theoretical Aspects of Software Engineering*. Ed. by Cristina David and Meng Sun. Cham: Springer Nature Switzerland, 2023, pp. 221–239. ISBN: 978-3-031-35257-7. DOI: 10.1007/978-3-031-35257-7_13. URL: http://dx.doi.org/10.1007/978-3-031-35257-7%5C_13.

[2] Elena Sherman and Matthew B. Dwyer. "Exploiting Domain and Program Structure to Synthesize Efficient and Precise Data Flow Analyses (T)". In: *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (Nov. 2015). DOI: 10.1109/ase.2015.41.

Common Variable Set Algorithm

Require: $V(\mathcal{I}_1) = V(\mathcal{I}_2) \wedge V(\Delta(\mathcal{I}_1, dv_1)) \subseteq V(\mathcal{I}_1) \wedge V(\Delta(\mathcal{I}_2, dv_2)) \subseteq V(\mathcal{I}_2)$
Ensure: $S_1 = S_2 \subseteq V(\mathcal{I}_1)$

function $\textsc{CommonVarSet}(dv_1,\ dv_2,\ \mathcal{I}_1,\ \mathcal{I}_2)$
  $S_1 \leftarrow \mathrm{V}(\Delta\,(\mathcal{I}_1, dv_1))$
  $S_2 \leftarrow \mathrm{V}(\Delta\,(\mathcal{I}_2, dv_2))$
  while $S_1 \neq S_2$ **do**
    if $S_1 \supset S_2$ **then**
      $dv_2 \leftarrow S_1 \setminus S_2$
      $S_2 \leftarrow S_2 \cup \mathrm{V}(\Delta\,(\mathcal{I}_2, dv_2)))$
    else if $S_2 \supset S_1$ **then**
      $dv_1 \leftarrow S_2 \setminus S_1$
      $S_1 \leftarrow S_1 \cup \mathrm{V}(\Delta\,(\mathcal{I}_1, dv_1)))$
    else if $S_1 \supset\subset S_2$ **then**
      $dv_1 \leftarrow S_2 \setminus S_1$
      $dv_2 \leftarrow S_1 \setminus S_2$
      $S_1 \leftarrow S_1 \cup \mathrm{V}(\Delta\,(\mathcal{I}_1, dv_1)))$
      $S_2 \leftarrow S_2 \cup \mathrm{V}(\Delta\,(\mathcal{I}_2, dv_2)))$
    end if
  end while
  return S_1
end function