Interprocedural Class Analysis[1]

Yoonseok Ko
Seoul National University
Programming Research Laboratory
2009. 1. 17.

[1]"A Framework for Call Graph Construction Algorithms", D. Grove and C. Chambers, ACM Trans. Program. Lang. Syst., 2001. 23(6):p. 685-746.

Agenda

- Preliminary
- Interprocedural Class Analysis
  - Flow graph representation
  - Edge filters
  - Call merging
  - Node merging
  - Merging parameters
- Instantiations
- Conclusions

Interprocedural Class Analysis

- **Framework integrates**
  - propagation-based analysis (0-CFA)
  - unification-based analysis
  - optimistic reachability analysis(RTA)
- **Computes set of classes for each program variable**

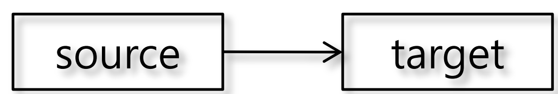
Flow graph representation

- Node for each variable, method, new, call
- Algorithm computes set of classes for each node

- Edge between two nodes if classes can flow between them

```
target = source;
```

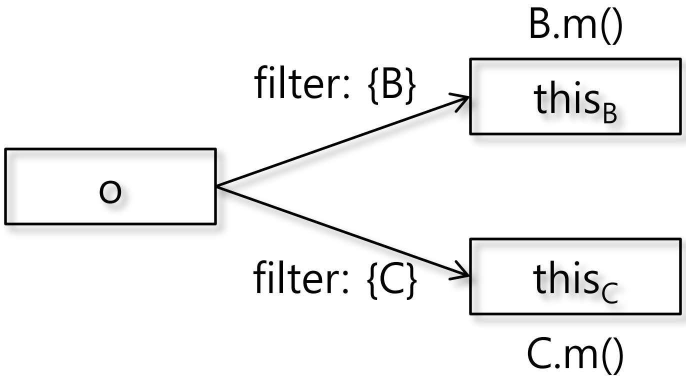
```
T1 m(T2 target) {…}
m(source);
```



Edge filters

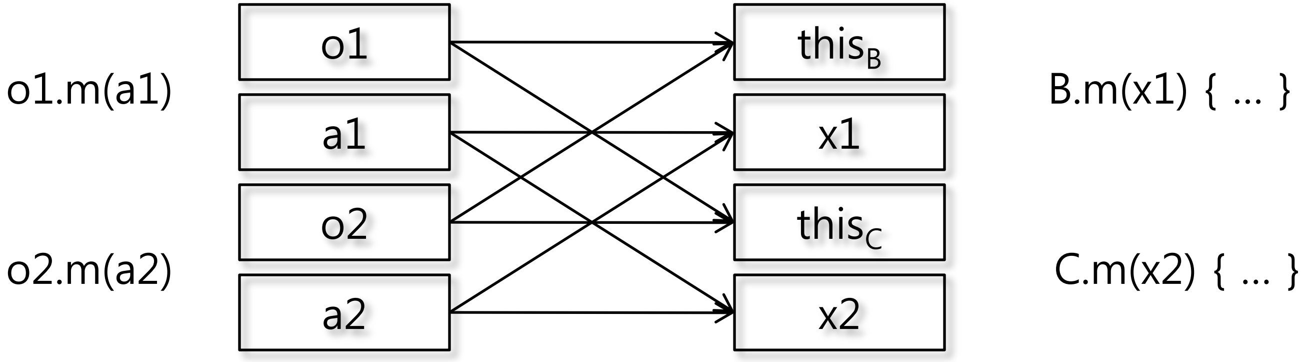
- Edges may have a filter set
- Don't propagate class if filter does not include that class

```
class B { m() { … this … }}
class C ext B { m() { … this … }}
B o = new C();
o.m()
```

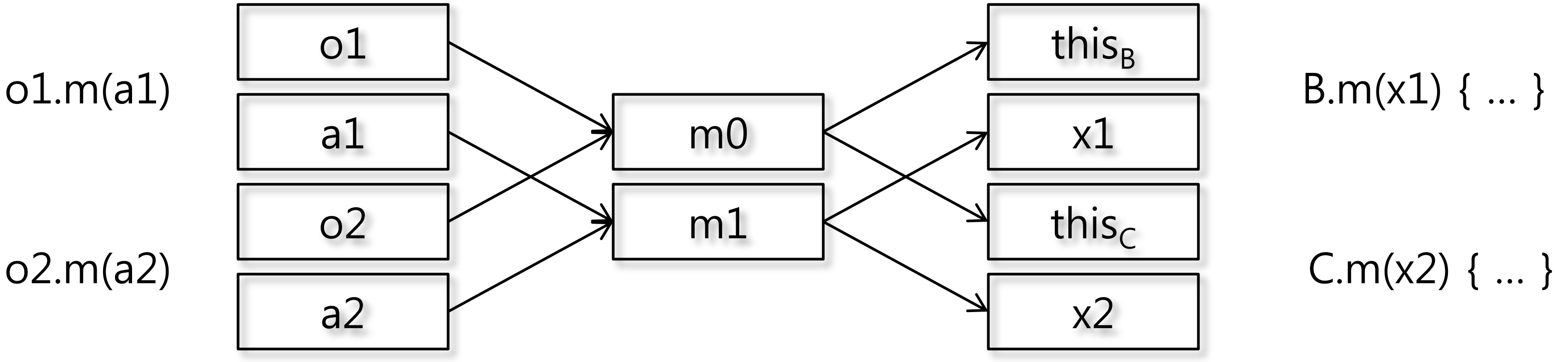


Call merging

- Analysis parameterized by MergeCalls
- When MergeCalls = false:



- When MergeCalls = true:

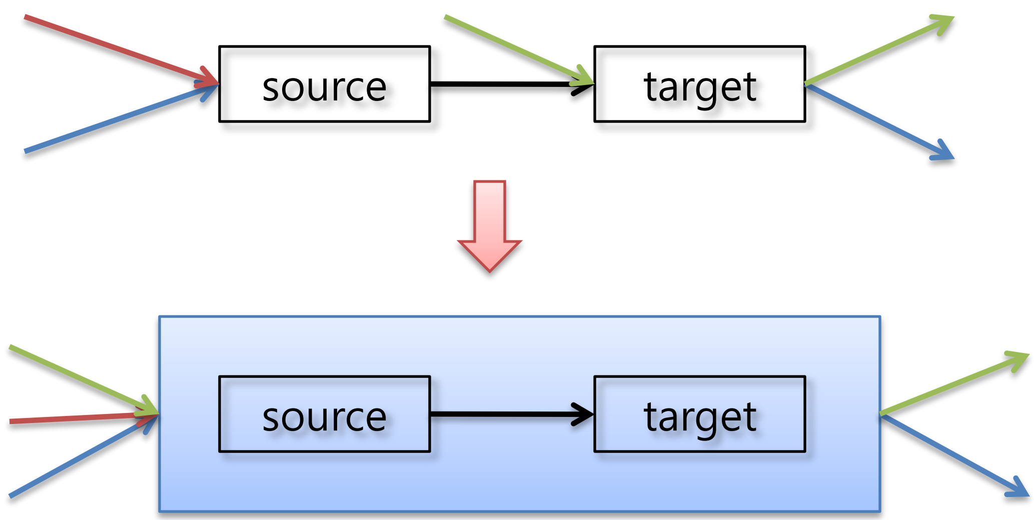


Node merging

- Can speedup analysis by merging nodes into supernodes
- Nodes merged with successors

```
target = source;

T1 m(T2 target) {…}
m(source);
```



- Always merging is equivalent to **unification-based analysis**

Merging parameters

- Analysis parameterized by
  - P and
  - MergeWithGlobal

- Parameters
  - P=k: merge node with its successors if node visited more than k times
  - P=0: always merge
  - P=N: never merge

  - MergeWithGlobal = true: use only one global supernode

Instantiations

Algorithm	P	MergeWithGlobal	MergeCalls	Complexity
0-CFA	N	N/A	false	$O(N^3)$
linear-edge 0-CFA	N	N/A	true	$O(N^2)$
Bounded 0-CFA	O(1)	false	false	$O(N^2$□$(N,N))$
Bounded linear-edge 0-CFA	O(1)	false	true	$O(N$□$(N,N))$
Simply bounded 0-CFA	O(1)	true	false	$O(N^2)$
Simple bounded linear-edge 0-CFA	O(1)	true	true	O(N)
Equivalence class analysis	0	false	true	$O(N$□$(N,N))$
RTA	0	true	true	O(N)

Conclusions

- **Parameterized algorithm**
  - from $O(N)$ to $O(N^3)$
- **Integrates both algorithm to achieve costs and precision benefits**
  - propagation-style analysis and
  - unification-style analysis

THANKS