# The Flow-Insensitive Precision of Andersen's Analysis in Practice

Sam Blackshear, **Bor-Yuh Evan Chang**, Sriram Sankaranarayanan

University of Colorado Boulder

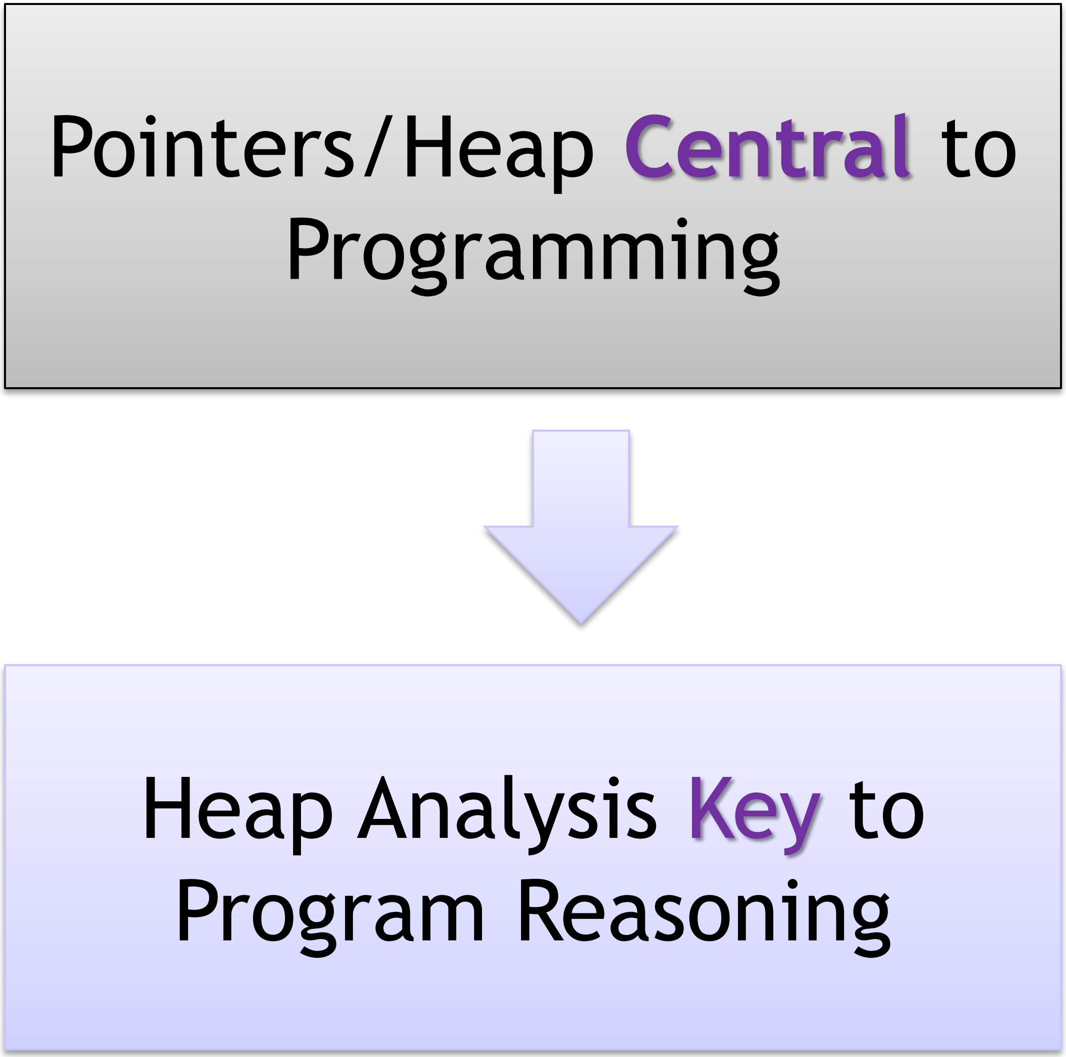
Manu Sridharan

IBM Research

UC Berkeley – June 10, 2011

Work to be presented at SAS 2011

# Pointers, pointers, pointers



Pointers/Heap **Central** to Programming

Heap Analysis **Key** to Program Reasoning

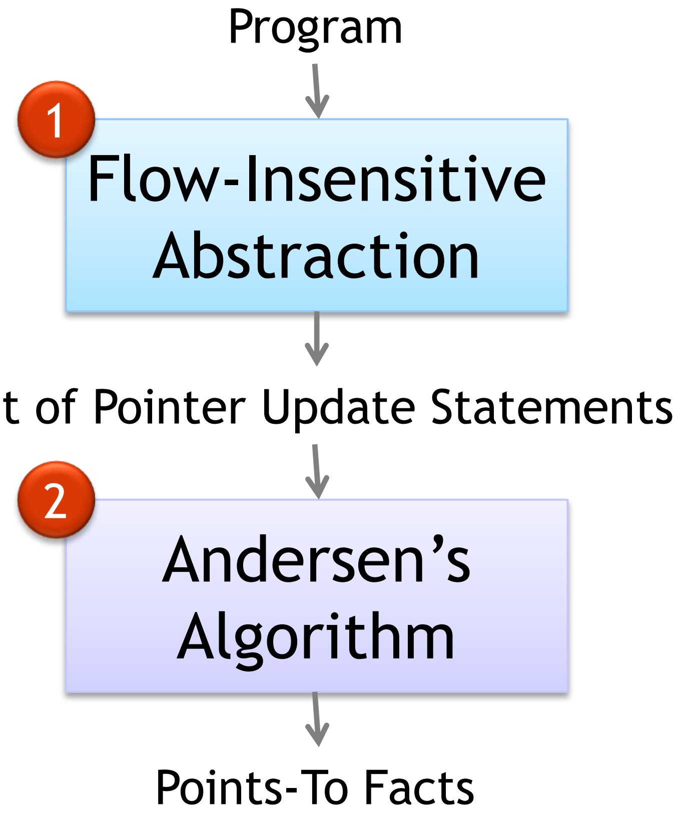
```
*p = q;    (C)

p.f = q;   (Java/C#/JS)
```

Property checkers (e.g., tainting, typestate, race conditions) are typical clients of pointer analysis.

# Never precise enough

- The Benchmark: Andersen’s Analysis
  - Sources of Imprecision?

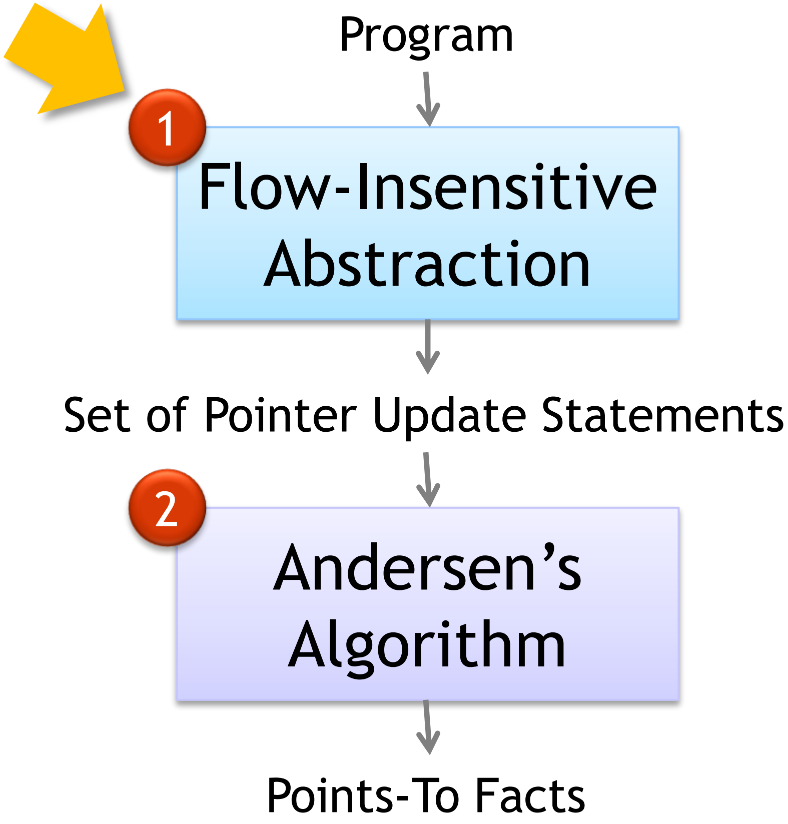

Program
1
Flow-Insensitive Abstraction
Set of Pointer Update Statements
2
Andersen’s Algorithm
Points-To Facts

Which should we attack?

Andersen’s is not a (fully) *precise flow-insensitive points-to analysis (PFIPTA)* [Chakaravarthy’03, Horwitz’97]

# Never precise enough

- The Benchmark: Andersen's Analysis
  - Sources of Imprecision?


Program
1
Flow-Insensitive Abstraction
Set of Pointer Update Statements
2
Andersen's Algorithm
Points-To Facts

Two Questions Arise:

*Theory*) Is there an efficient algorithm for precise fl tive analysis?

This Talk

*Practice*) Is there a precision gap with Andersen's in practice?

No

# Answering “precision in practice”

- An algorithm for precise flow-insensitive points-to analysis (for finite memory)
  - based on an on-demand witness search algo.
  - with a SAT encoding, “efficient enough” for experimentation

- Ask experimentally: Is an Andersen’s derived-fact ever refuted by our precise algorithm?

# Roadmap

- Background: Imprecision in Andersen’s

2 Andersen’s Algorithm

- Precise Analysis by Witness Search

- Experimental Findings: Is There a Precision Gap in Practice?

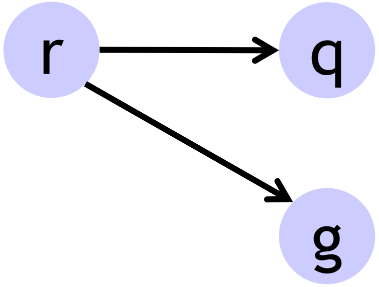
# The Points-To Analysis Problem

Given a set of assignments of the form

$*^n$ p := &q; $*^n$ p := $*^m$ q; finite memory

$*^n$ p := malloc(); with dynamic memory

Compute a (may) points-to graph



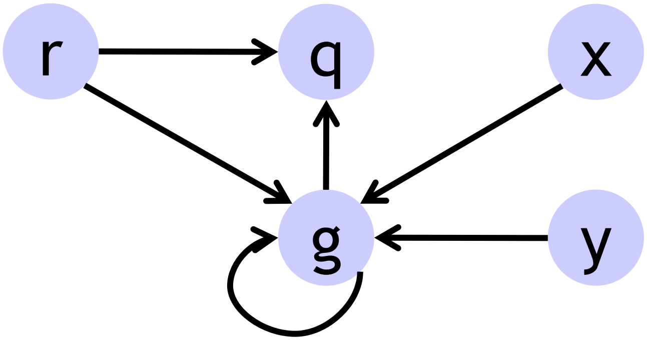
“r may sometime contain the address of q”

“r may point to g”

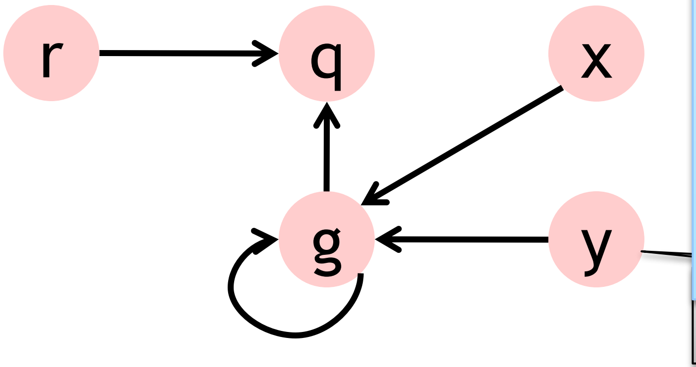
abstract location modeling one or more concrete cells

# Precise Flow-Insensitive Points-To Analysis

Andersen's analysis



Exact graphs and an



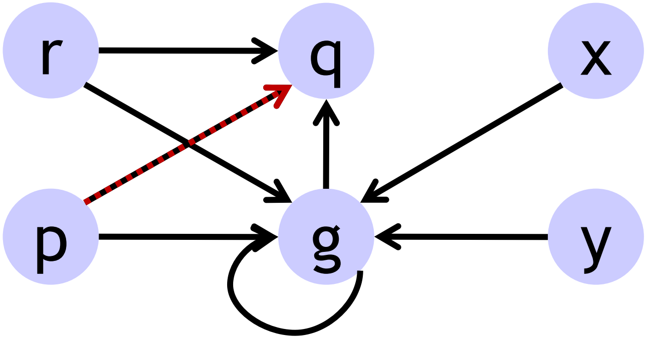
models a single cell

An edge is **realizable** iff it is in an exact graph after some seq. of updates (from empty)

A precise flow-insensitive points-to analysis

- derives all realizable edges and no others
- i.e., derives a precise join of all exact graphs along all possible executions

# Imprecision: Simultaneous Points-To



p := *r;

**Unrealizable!**

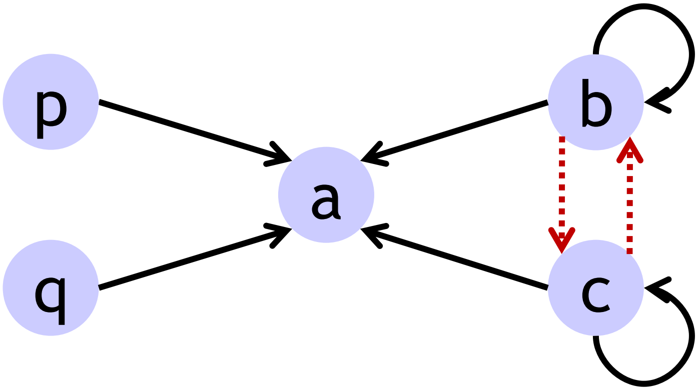
Requires simultaneously

r → q and r → g

or simultaneously

g → g and g → q

# Imprecision: Decomposing Multi-Derefs



$**p := *q;$

**Unrealizable!**

But realizable with

$t_1 := *p;\ t_2 := *q;\ *t_1 := t_2;$

# Roadmap

- Background: Imprecision in Andersen’s

2 Andersen’s Algorithm

- Precise Analysis by Witness Search

- Experimental Findings: Is There a Precision Gap in Practice?

# Witnesses

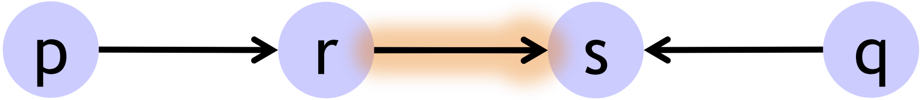
A witness for an edge e is an execution (or, a sequence of assignments)

$$\{\} \xrightarrow{a_1} G_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} G_n$$

where $e \in G_n$

Idea: Given an edge e to witness, search backwards over possible executions constrained by the initial analysis

# Edge Dependency Rules



*p := q;

## Dependency Rule

$$r \rightarrow s \xLeftarrow{*p := q} \{ p \rightarrow r, q \rightarrow s \}$$

is realizable

if

are simultaneously realizable

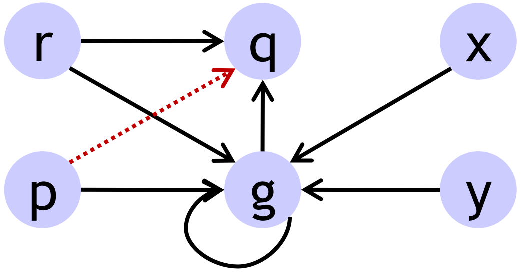
# Search by rewriting using dependency rules

$$r \to g \xLeftarrow{r := {*}x} \{ x \to g,\ g \to g \}$$

$$\xLeftarrow{{*}x := y} \{ x \to g,\ y \to g \}$$

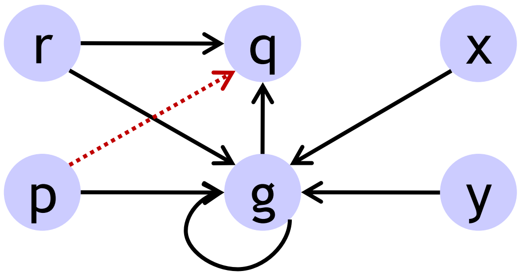
$$\xLeftarrow{y := x} \{ x \to g \}$$

$$\xLeftarrow{x := \&g} \{ \}$$



# Refutation yields precision improvement

$$p \to q \xLeftarrow{p\ :=\ *r} \{\ r \to g,\ g \to q\ \}$$



$$\cancel{\xLeftarrow{*x\ :=\ r} \{\ r \to g,\ x \to g,\ r \to g\ \}}$$

$$\cancel{\xLeftarrow{r\ :=\ *x} \{\ x \to g,\ g \to g,\ g \to q\ \}}$$

Proven

**Unrealizable!**

# Roadmap

- Background: Imprecision in Andersen's

2 Andersen's Algorithm

- Precise Analysis by Witness Search

- Experimental Findings: Is There a Precision Gap in Practice?

# Evaluation Methodology Overview

Is there a precision gap in practice?

Is there a witness for every points-to fact derived by Andersen’s? Yes ⇒ No Gap

Test Configurations

- Factor out imprecision due to dynamic memory (summary nodes)
- Factor out imprecision due to decomposing multi-dereferences
- What about for alias queries? $\exists r.\{p \rightarrow r, q \rightarrow r\}$?

# Summary Nodes and Dynamic Memory

Standard Practice: structs, arrays, malloc modeled by summary nodes

– than one concrete cell

Decidability of precise flow-insensitive points-to analysis with dynamic memory allocation is unknown

Bounding the Precision Gap with Summaries

- Lo du
- Upper: Treat summaries as abstracting one concrete cell (under-approx. analysis)

Always find witnesses = No precison gap!

(factoring out decomposing multi-dereferences)

# Evaluation Benchmarks

| | program size | problem size | lower bound | | upper bound | |
|---|---|---|---|---|---|---|
| | kloc | num pt edges | depth | time (s) | depth | time (s) |
| aget | | | | | | |
| arp | | | | | | |
| slattach | | | | | | |
| netstat | | | | | | |
| ifconfig | | | | | | |
| stunnel | 17.1 | 426 | | | | |
| plip | 18.4 | 1052 | | | | |
| knot | 1.3 | 29 | | | | |
| esp | 10.9 | 637 | | | | |
| ide-disk | 12.6 | 437 | | | | |
| bc | 6.2 | 453 | 7.2 | 10.6 | 7.2 | 88.9 |
| watchdog | 9.4 | 1027 | 6.3 | 2698.3 | 6.5 | 4982.0 |

Feasability:
Small search depths

12 benchmarks
(small- to medium-sized in C)

over 4 categories
(network utilities, device drivers, terminal application, system daemon)

# Decomposing Multi-Derefs and Aliasing

## Decomposing Multi-Dereferences

- Witness search over transformed statements
- Post-pass to validate w.r.t. original statements
- All witnesses validate for lower bound config. and 97.5% (4561/4676) for upper bound config.
  - Definitely no gap factoring out summaries imprecision
  - At most tiny gap considering summaries imprecision

## Alias Queries

- Witness search on 1000 random pairs of vars
- Always found witnesses ⇒ No observed gap!

# Conclusion

- Empirically Observed: No (or ≤tiny) gap between Andersen's and PFIPTA
  - Witnesses are short

- Target Imprecision from Flow-Insensitivity
  - Witness refutation with aspects of flow-sensitivity
  - Get on-demand refinement with flow-sensitivity

1 Flow-Insensitive Abstraction