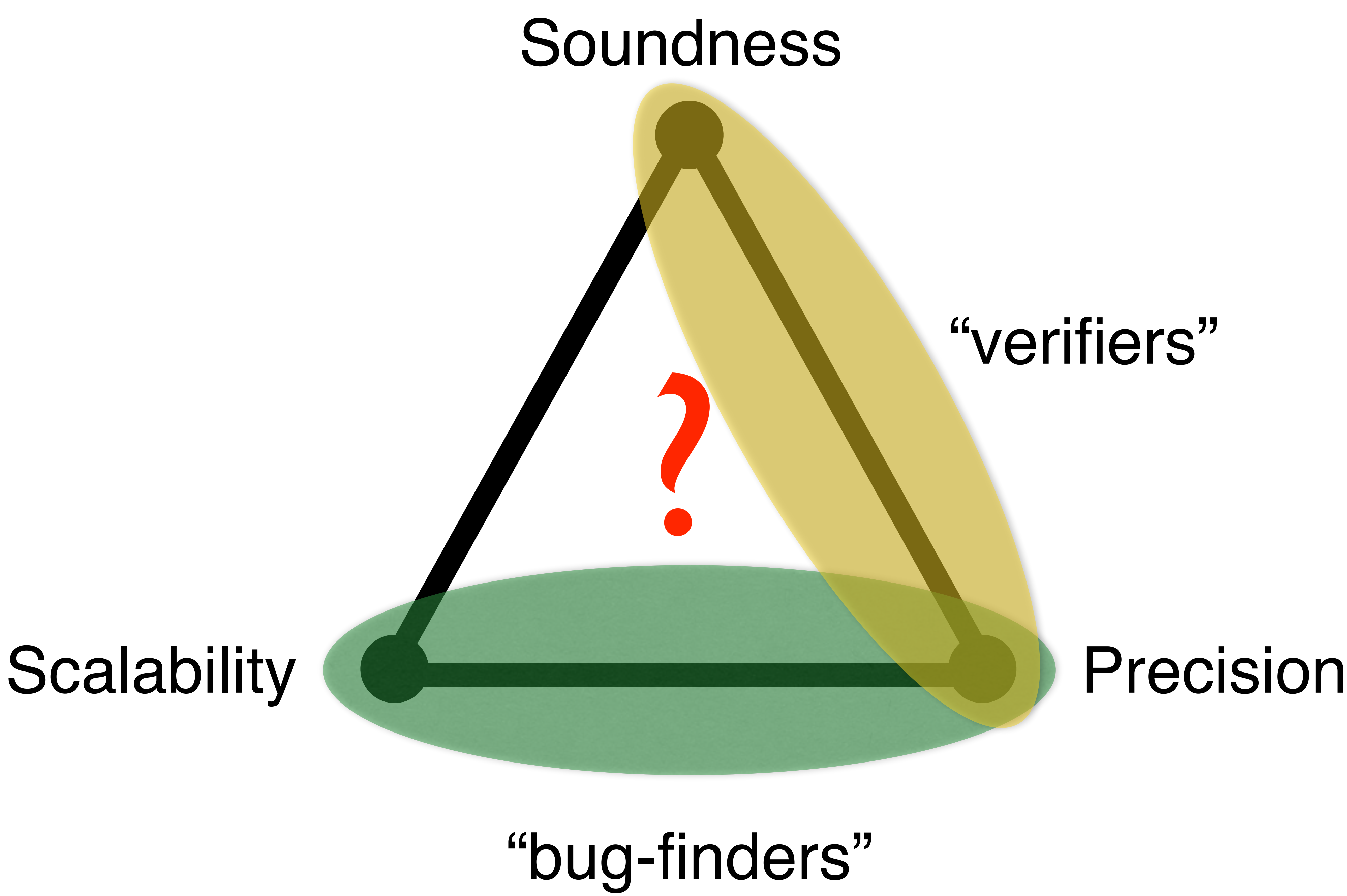
Selective Context-Sensitivity Guided by Impact Pre-Analysis

Hakjoo Oh[1] Wonchan Lee[1] Kihong Heo[1]
Hongseok Yang[2] Kwangkeun Yi[1]

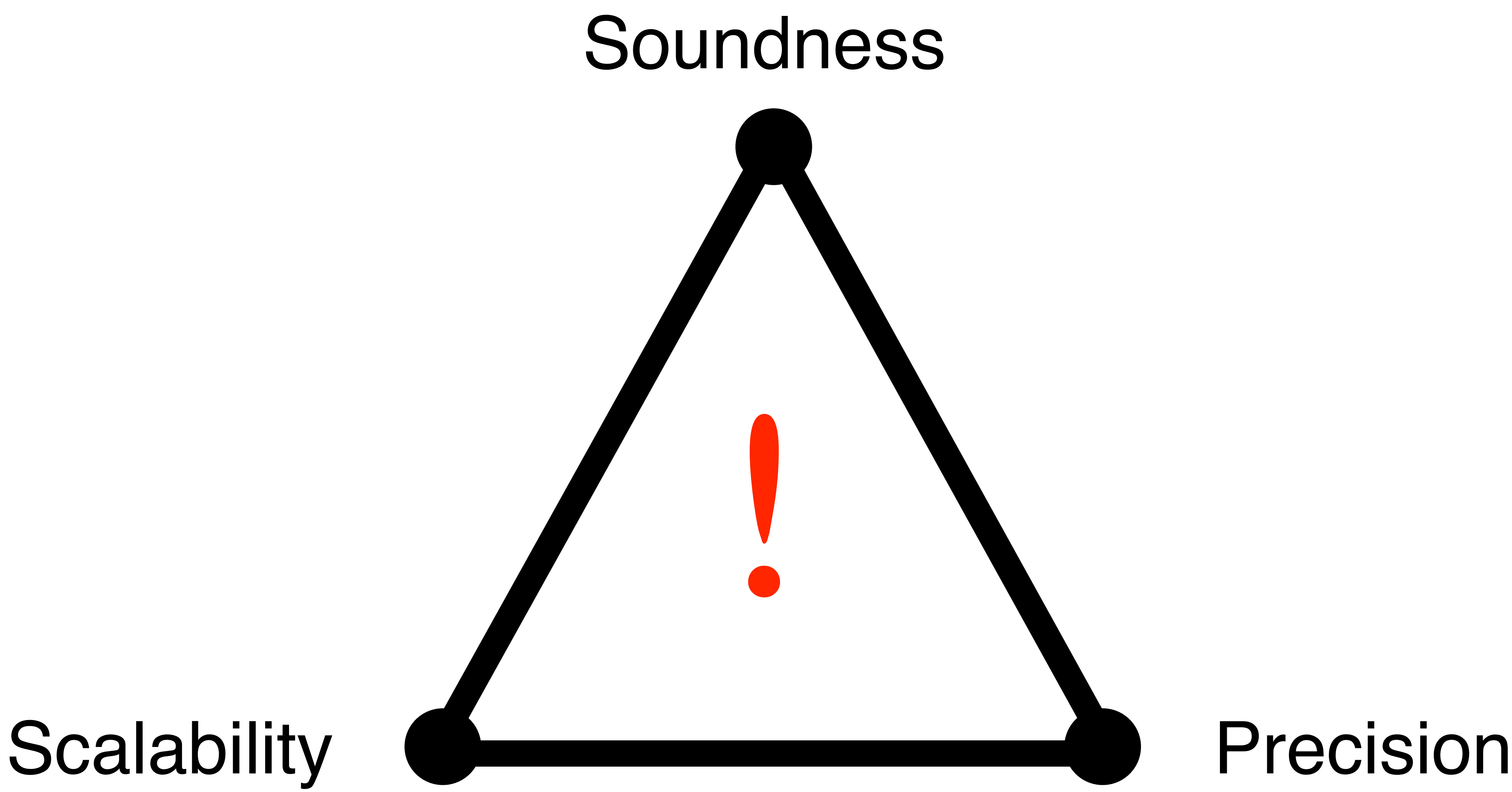
[1]Seoul National University
[2]University of Oxford

PLDI 2014 @Edinburgh, Scotland

Challenge in Static Analysis



Our Long-term Goal



General Sparse Analysis Framework [PLDI'12]

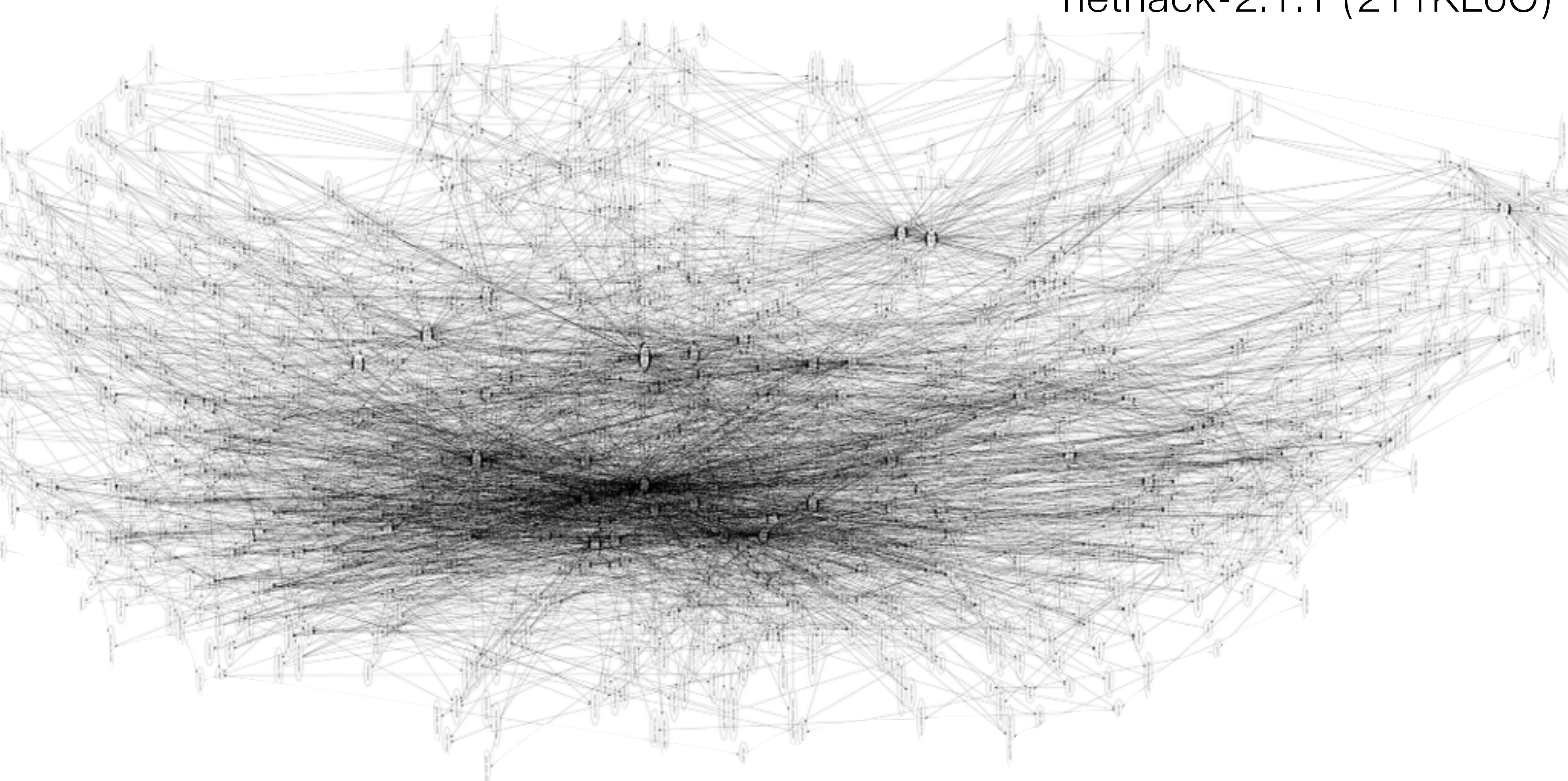
Selective X-Sensitivity Approach [PLDI'14]

Motivation

- In 2007, we commercialized Sparrow The Early Bird
  - memory-bug-finding tool for full C
  - designed in abstract interpretation framework
  - sound in design, unsound yet scalable in reality
- Realistic workbench available
  - “let’s try to achieve sound, precise, yet scalable version”

The Challenge in Reality

nethack-2.1.1 (211KLoC)

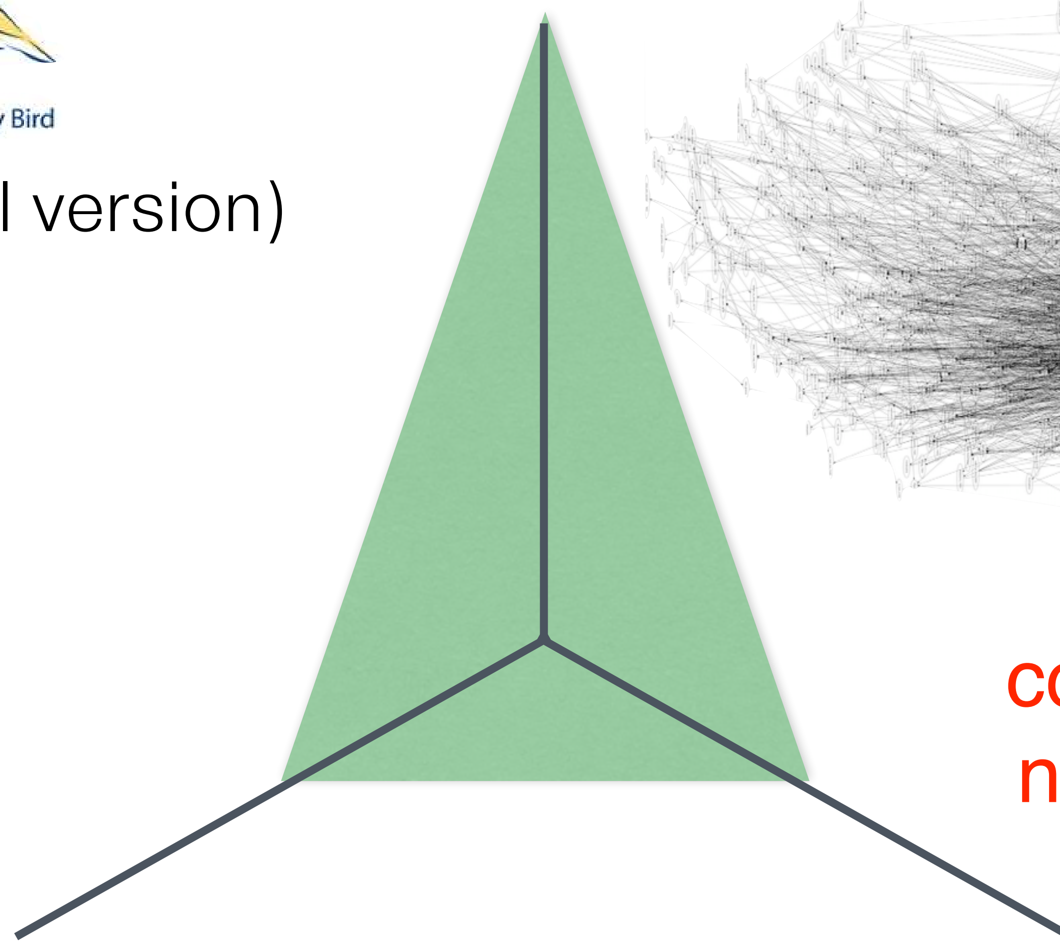


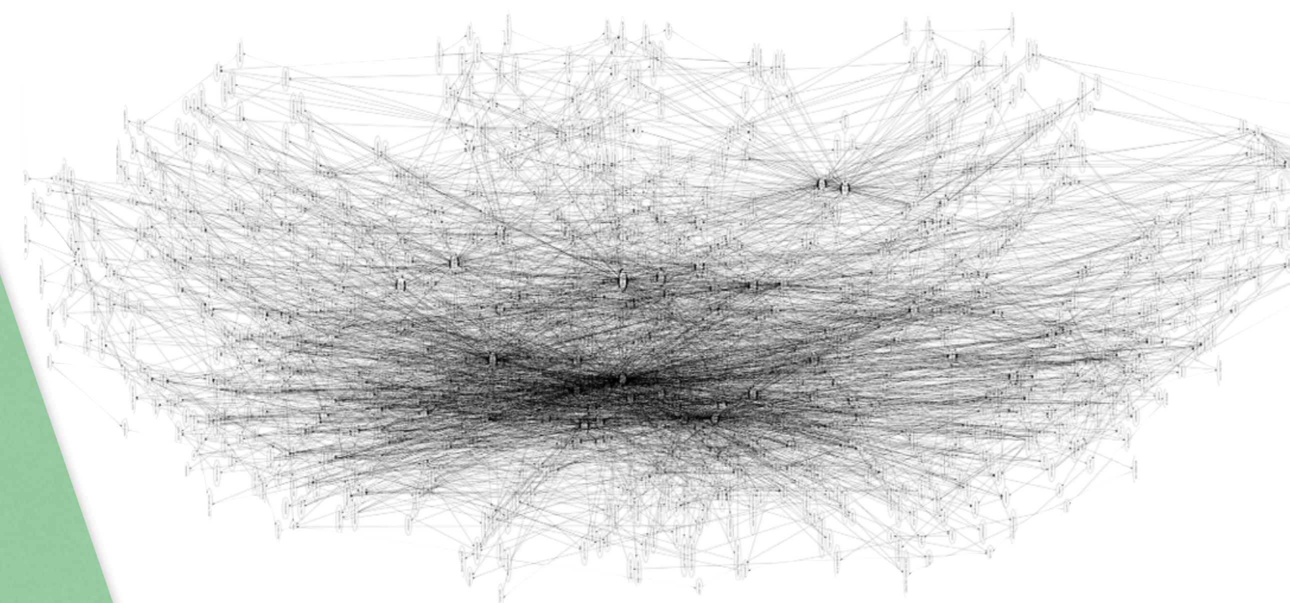
The Challenge in Reality

Soundness

Sparrow
The Early Bird

(2007, sound-&-global version)





35KLoC

context-insensitive
non-relational, etc

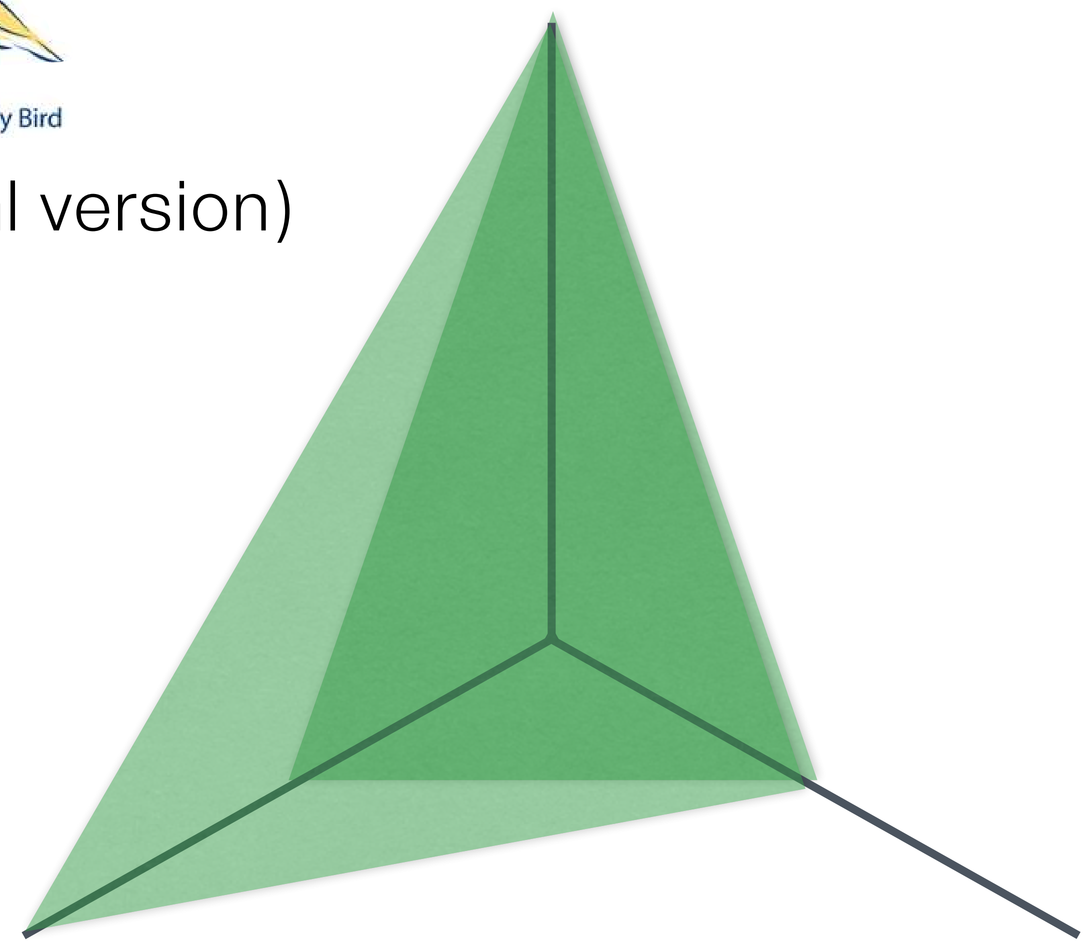
Scalability

Precision

Scalability, Done.

Sparrow
The Early Bird

(2012, sound-&-global version)



General Sparse Analysis Framework [PLDI'12]

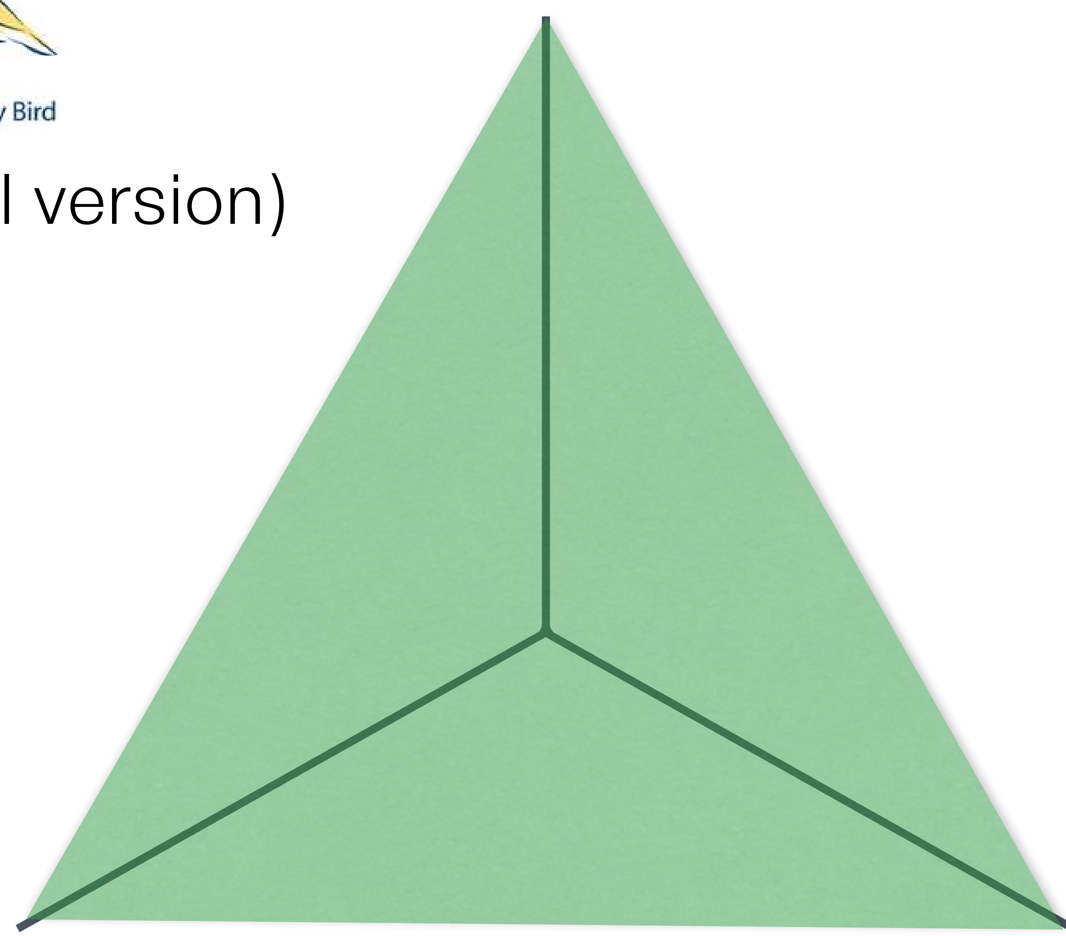
This Paper

The Second Goal: Precision

Soundness

Sparrow
The Early Bird

(2014, sound-&-global version)



1 Million LoC

context-sensitivity,
relational analysis

Scalability

General Sparse Analysis Framework [PLDI'12]

Precision

Selective X-Sensitivity Approach [PLDI'14]

Effectiveness for Context-Sensitivity

24% / 28%

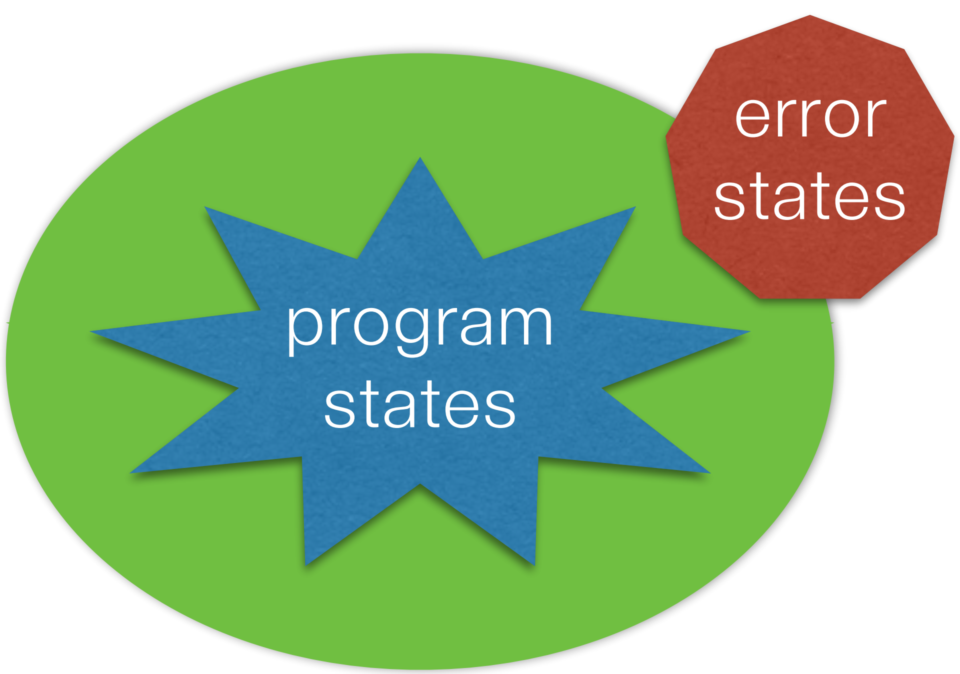
Reduction of false alarms

Increase of analysis time

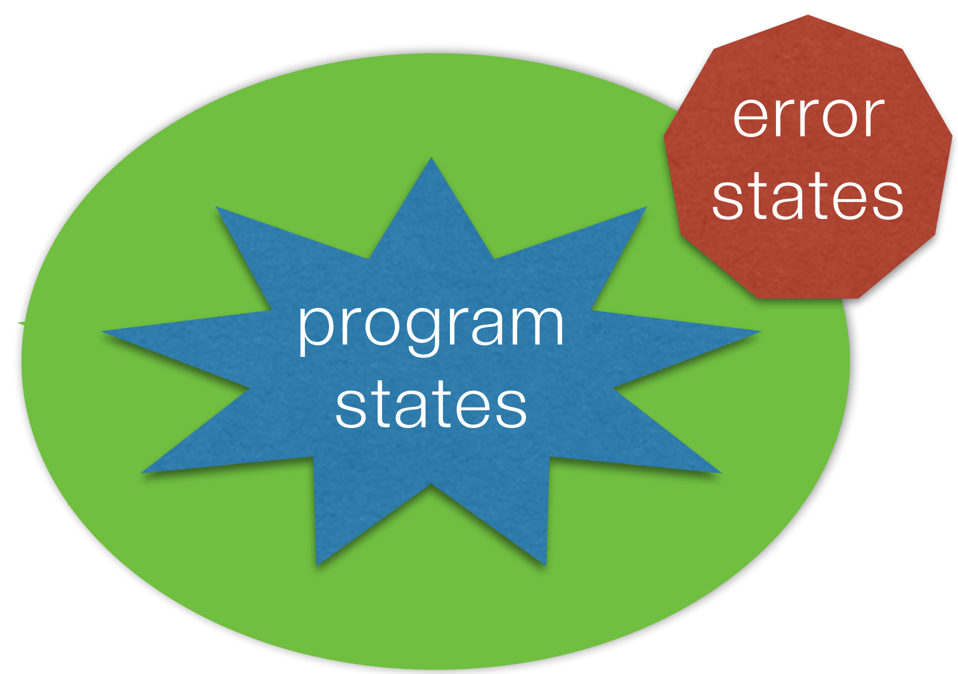
vs. context-insensitivity

Selective X-Sensitivity

- Apply precision(X) only when/where it matters
- X = context-sensitivity, relational analysis, etc



vs.



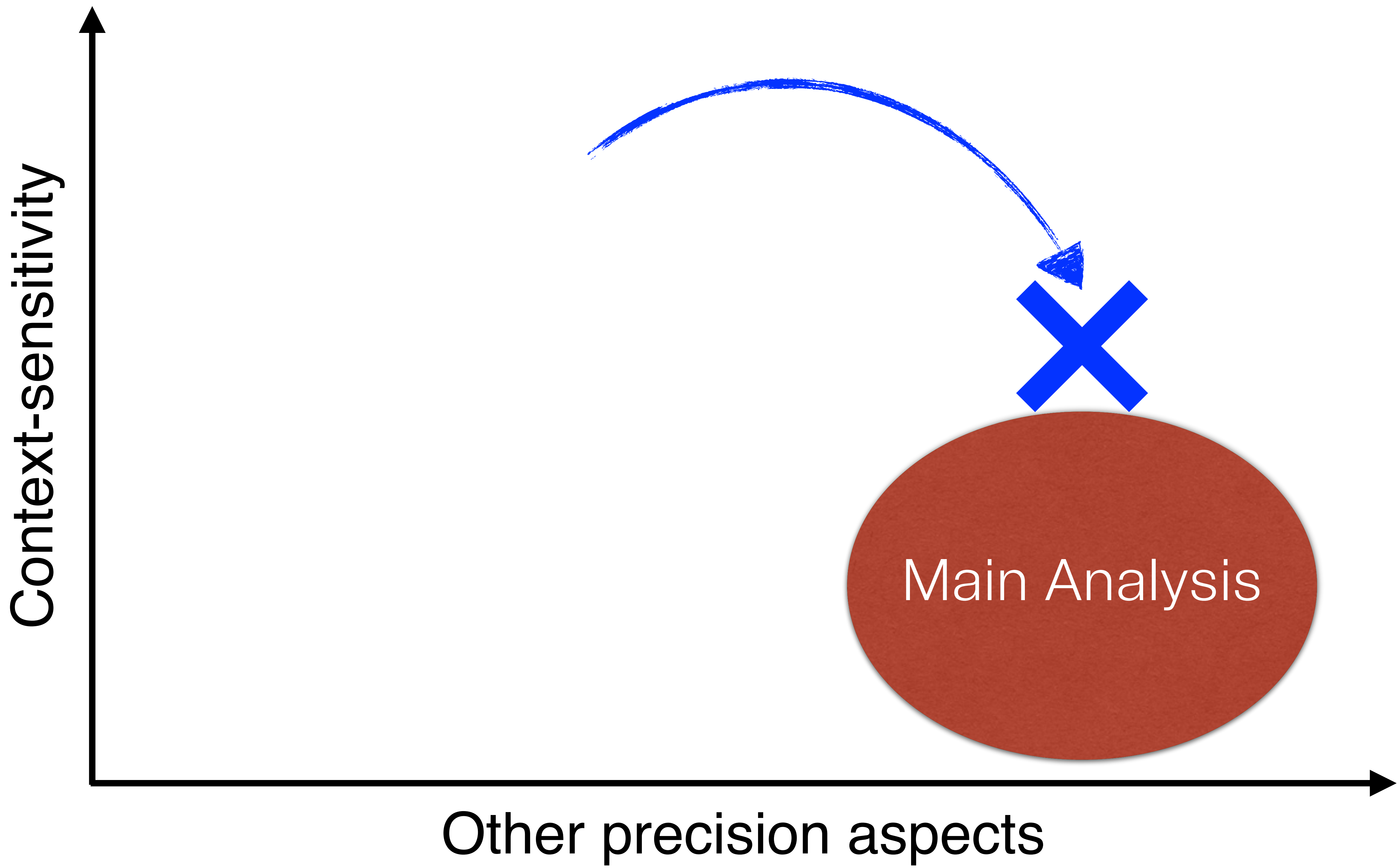
our method: **24%** / **28%**

3-CFA: **24%** / **1300%**

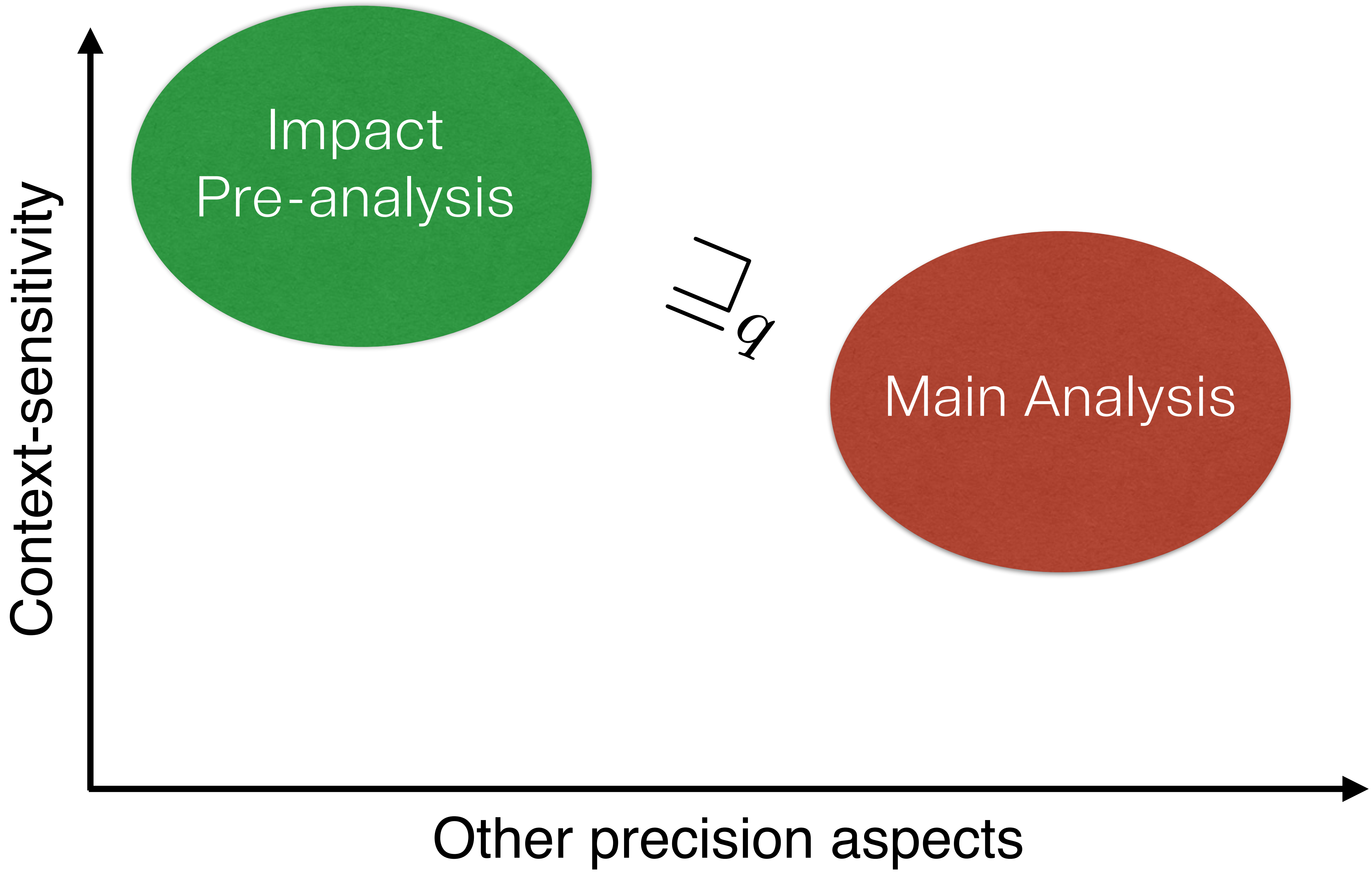
Key Idea: Impact Pre-Analysis

- Estimate the impact of X-sensitivity on main analysis
  - fully X-sensitive
  - but, approximated in other precision aspects

Key Idea: Impact Pre-Analysis



Impact Realization



Two Instance Analyses

- Selective context-sensitivity
- Selective relational analysis

Selective Context-Sensitivity

Example Program

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);   // Q1        always holds
c2:   y = h(input());
      assert(y > 1);   // Q2        does not always hold
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```

Context-Insensitivity

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);   // Q1
c2:   y = h(input());
      assert(y > 1);   // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```

[-∞,+∞]

Context-insensitive interval analysis cannot prove Q1

Context-Sensitivity: 3-CFA

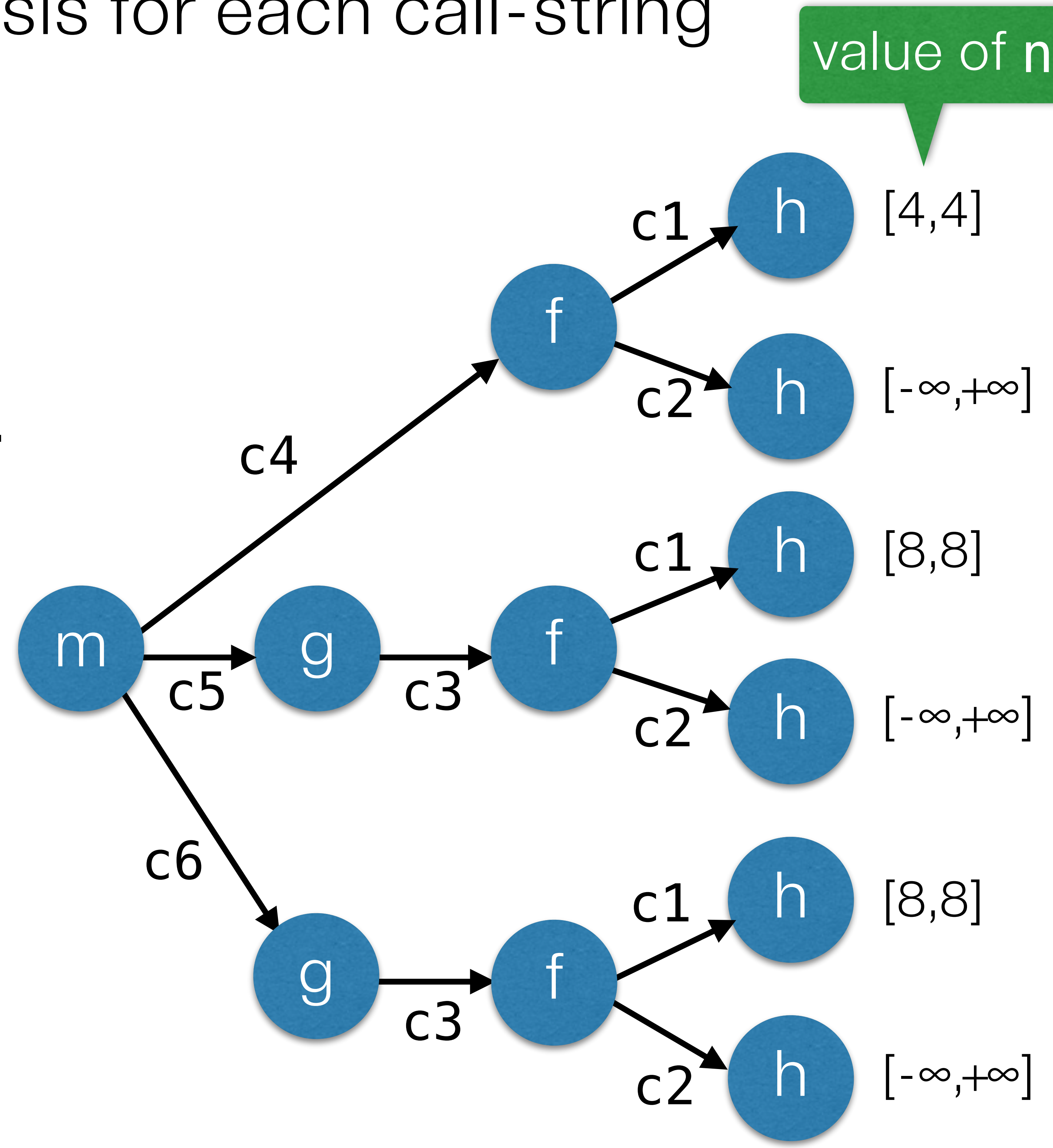
Separate analysis for each call-string

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



Context-Sensitivity: 3-CFA

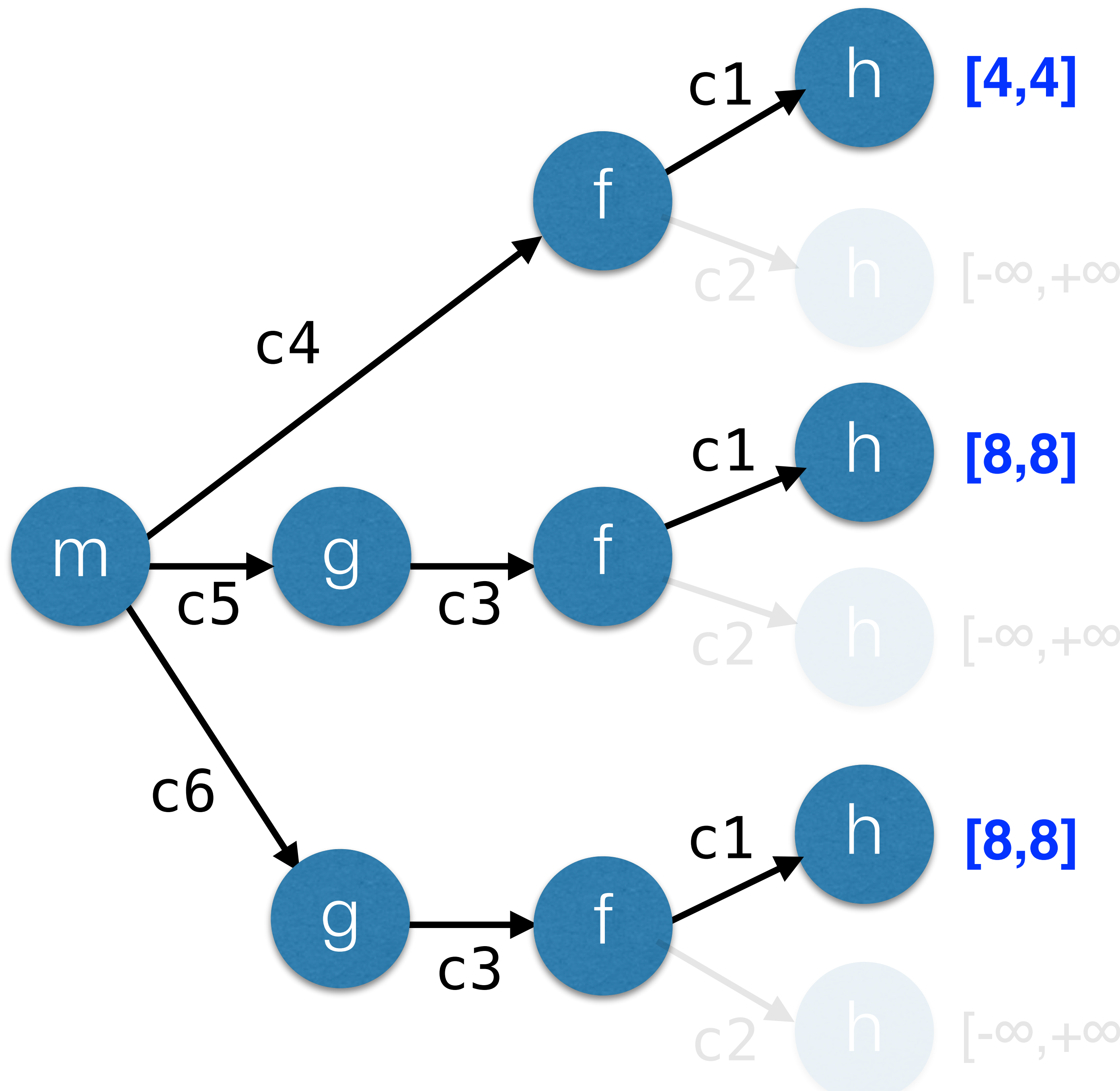
Separate analysis for each call-string

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



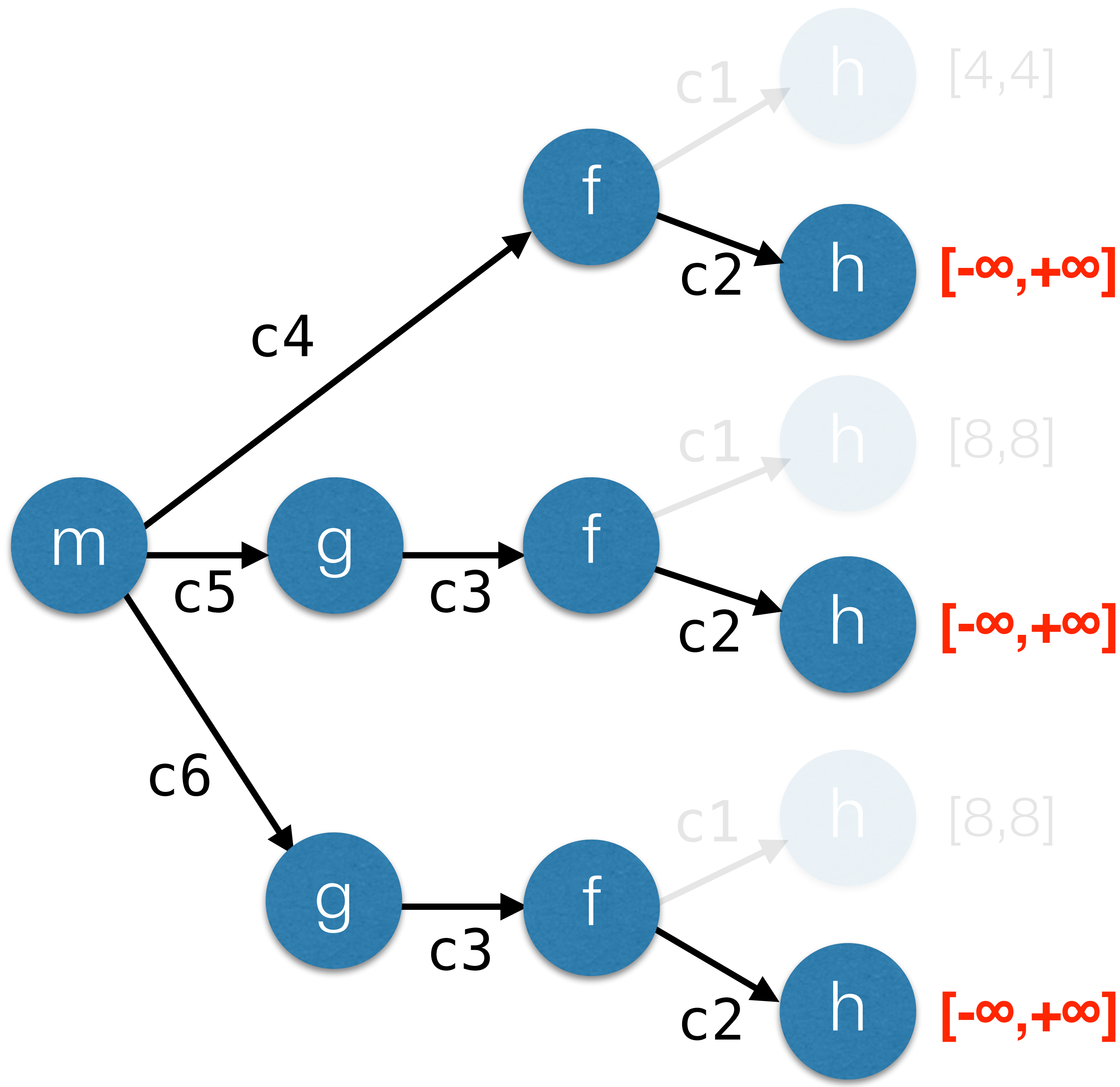
Problems of k-CFA

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



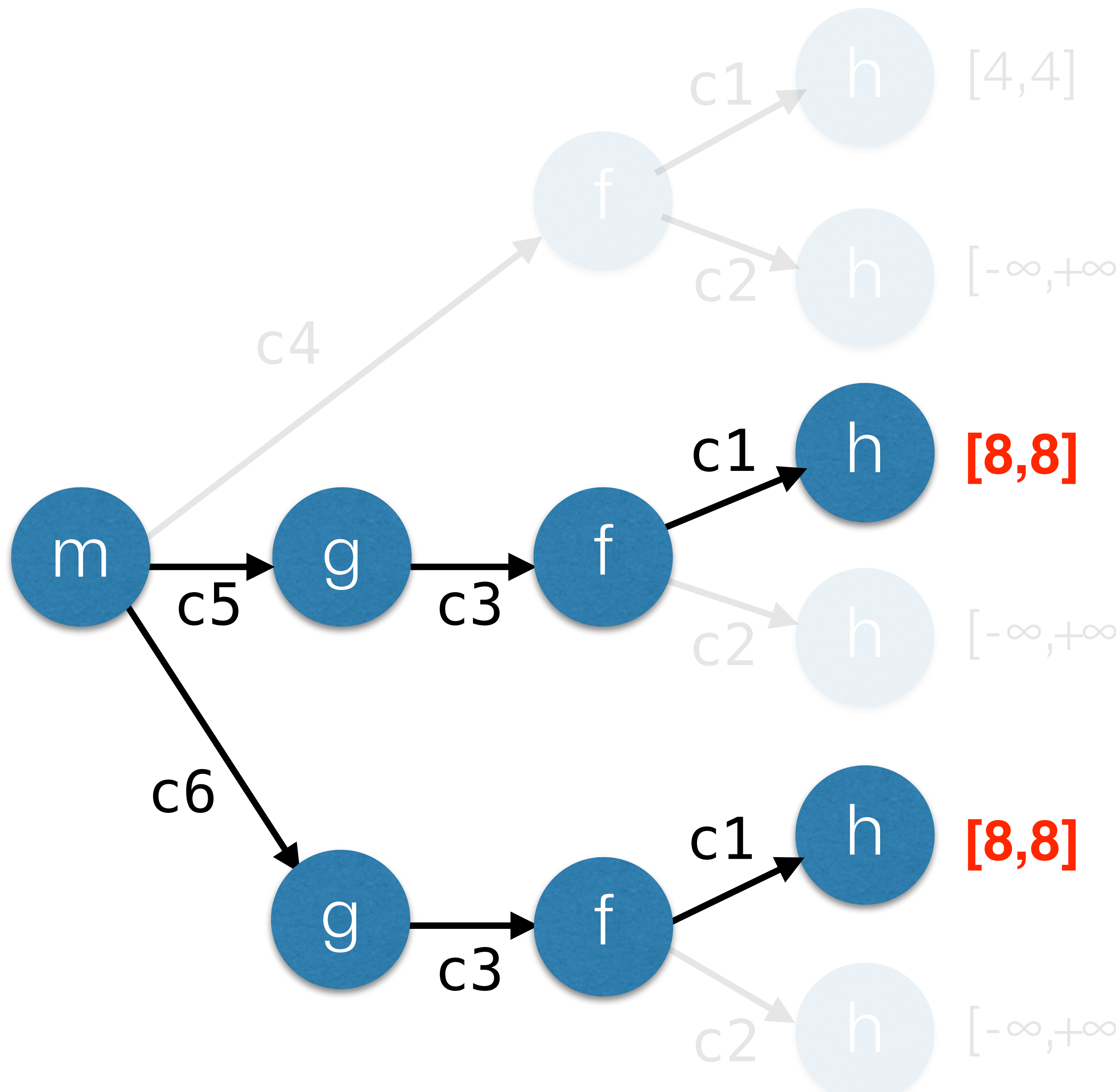
Problems of k-CFA

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



Our Selective Context-Sensitivity

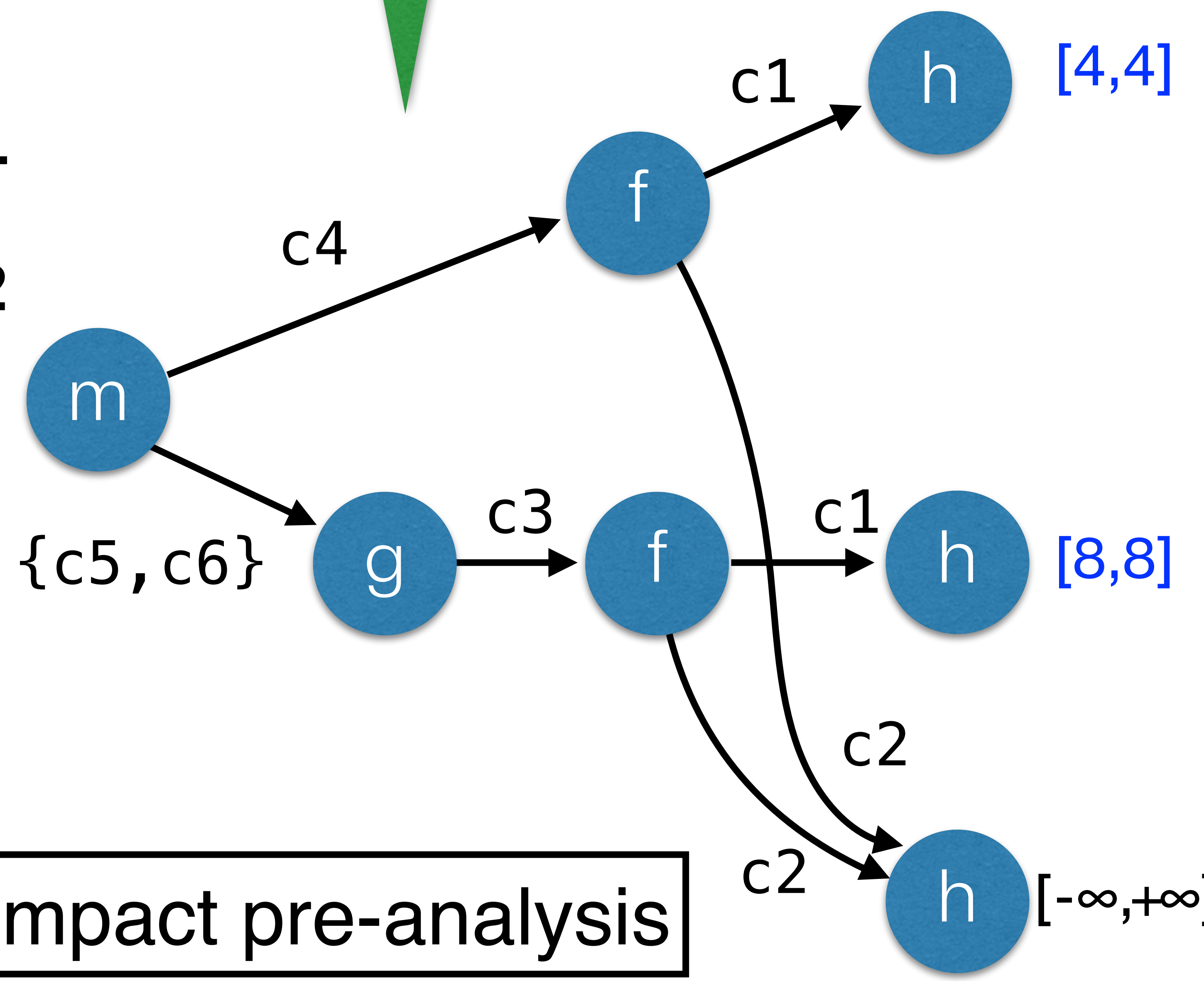
```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```

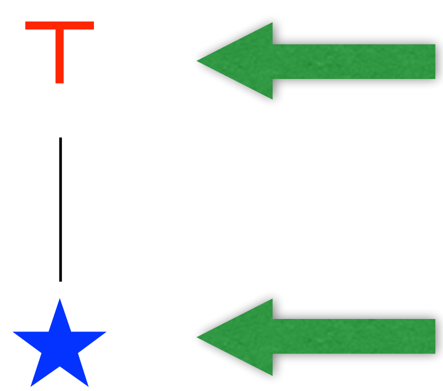
Challenge: How to infer this selective context-sensitivity?



Our solution: Impact pre-analysis

Impact Pre-Analysis

- Full context-sensitivity
- Approximate the interval domain



$\top$ all intervals

non-negative intervals, e.g., [5,7], [0,∞]

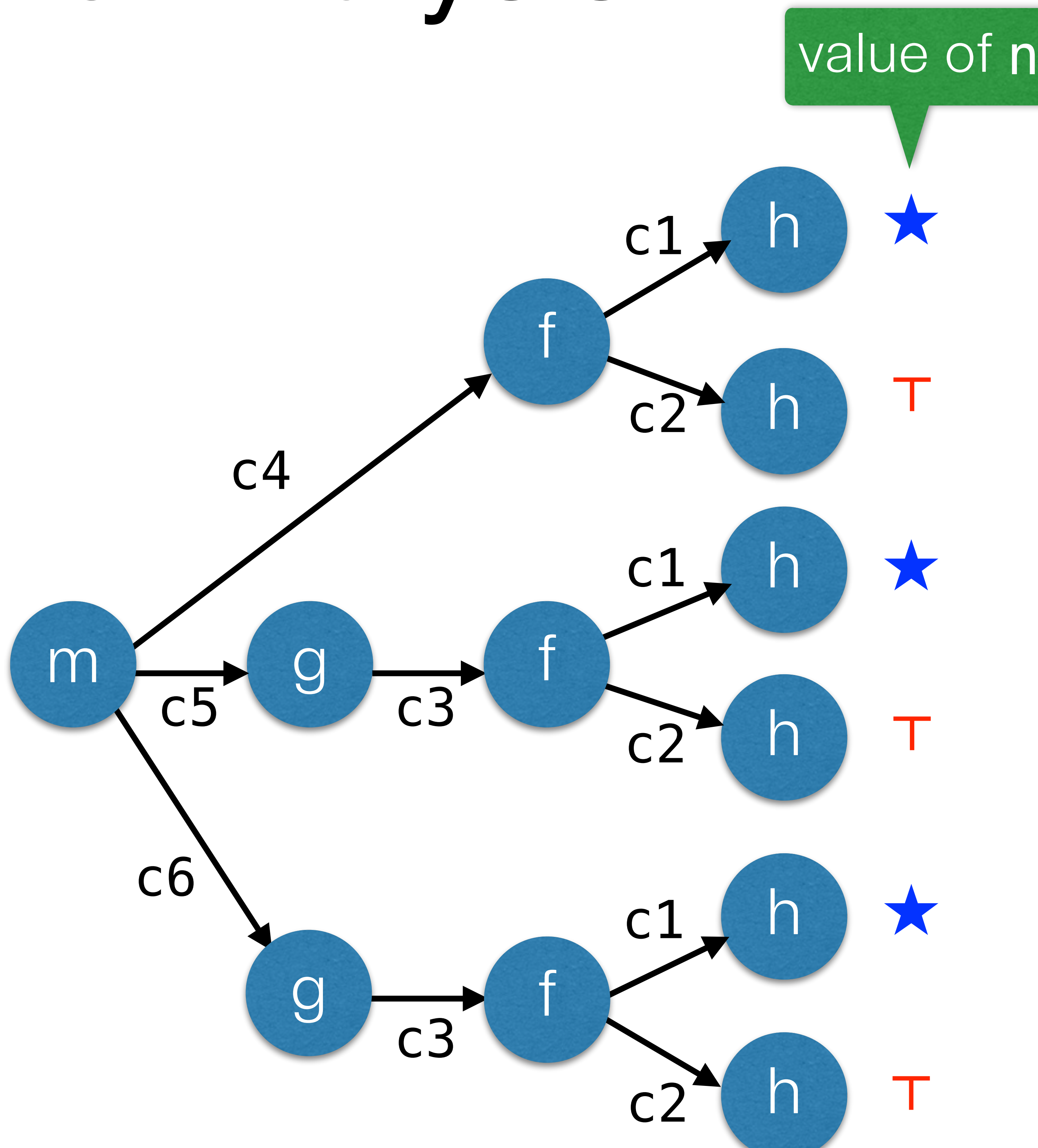
Impact Pre-Analysis

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



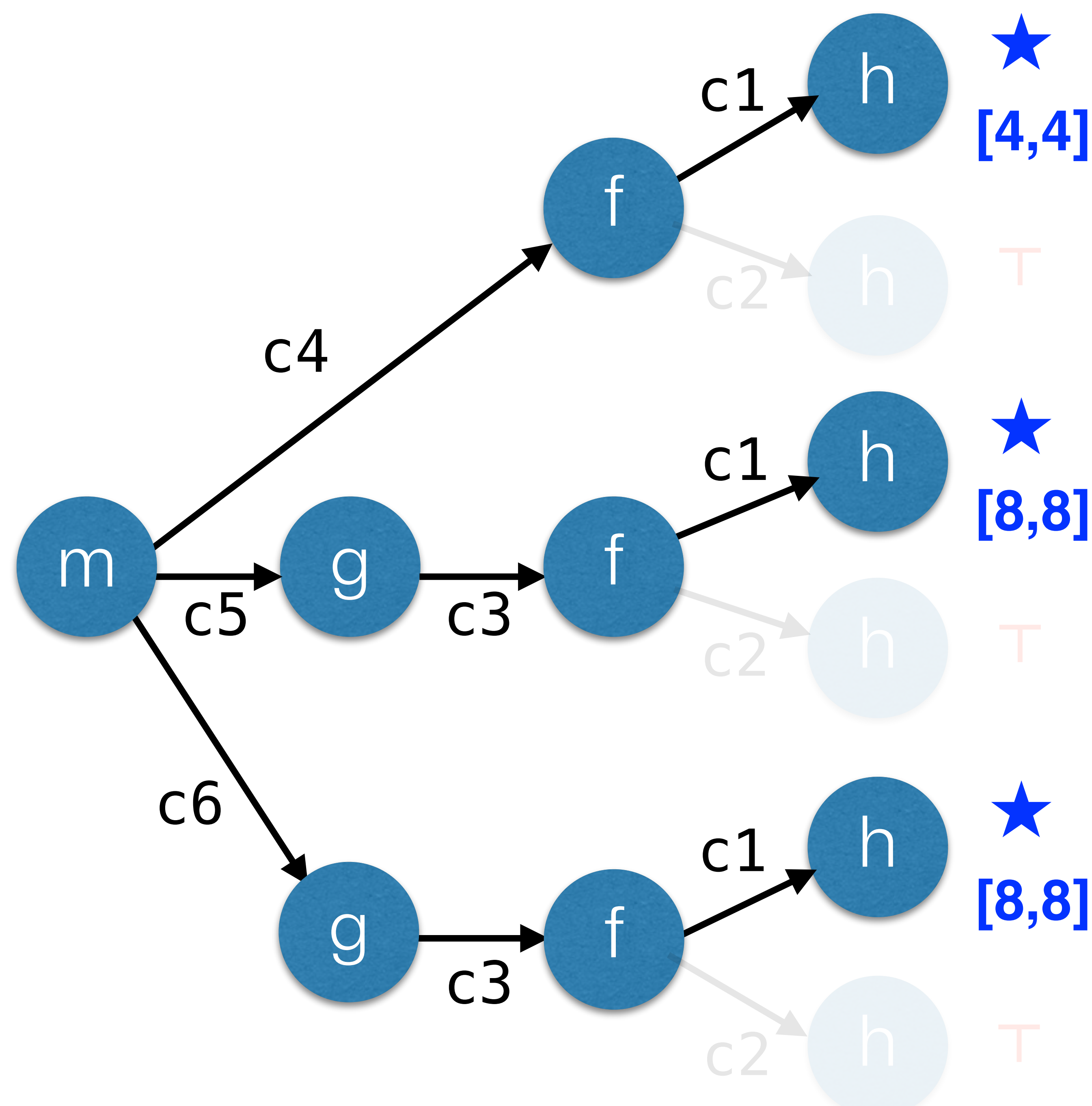
Impact Pre-Analysis

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



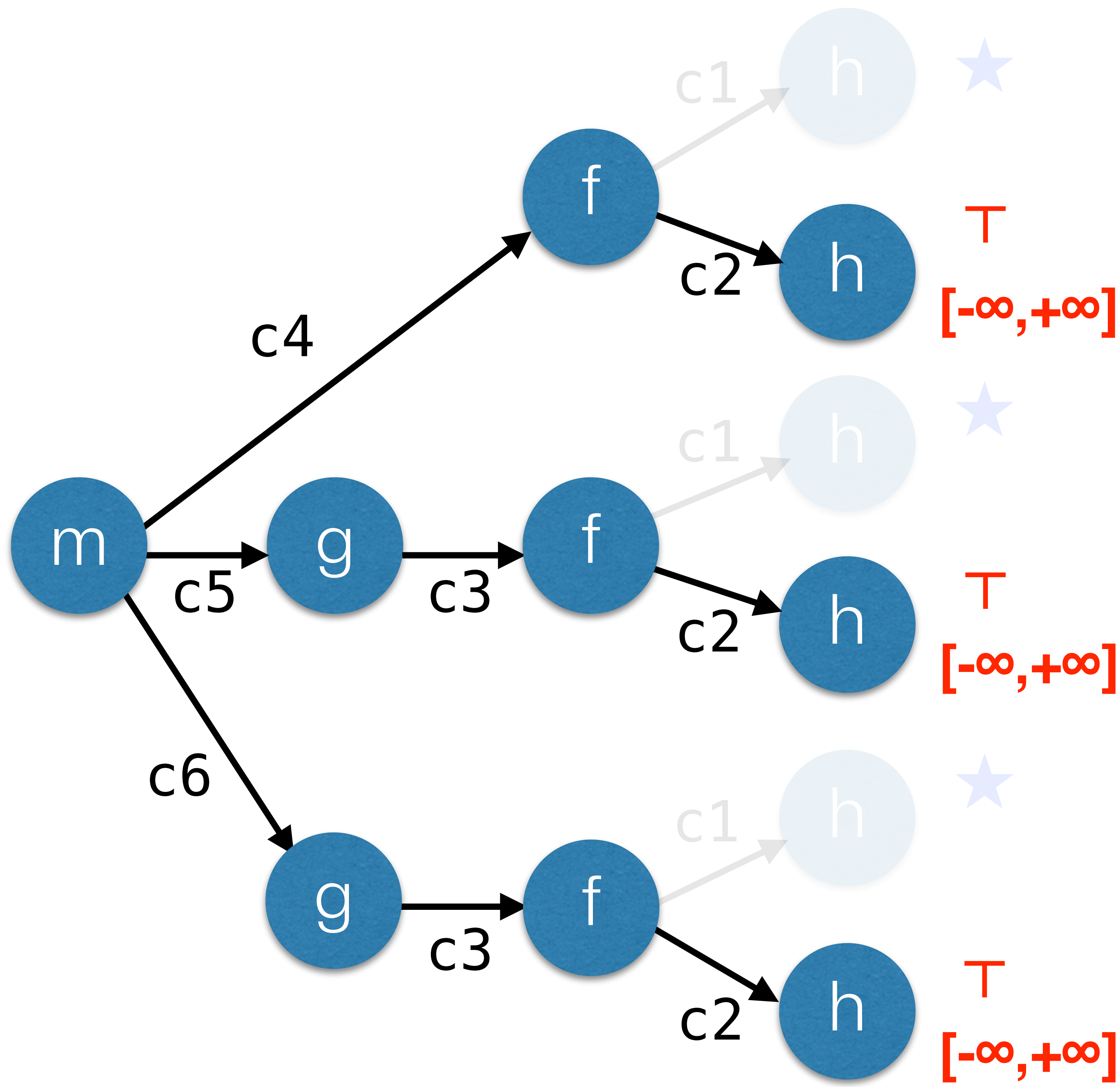
Impact Pre-Analysis

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



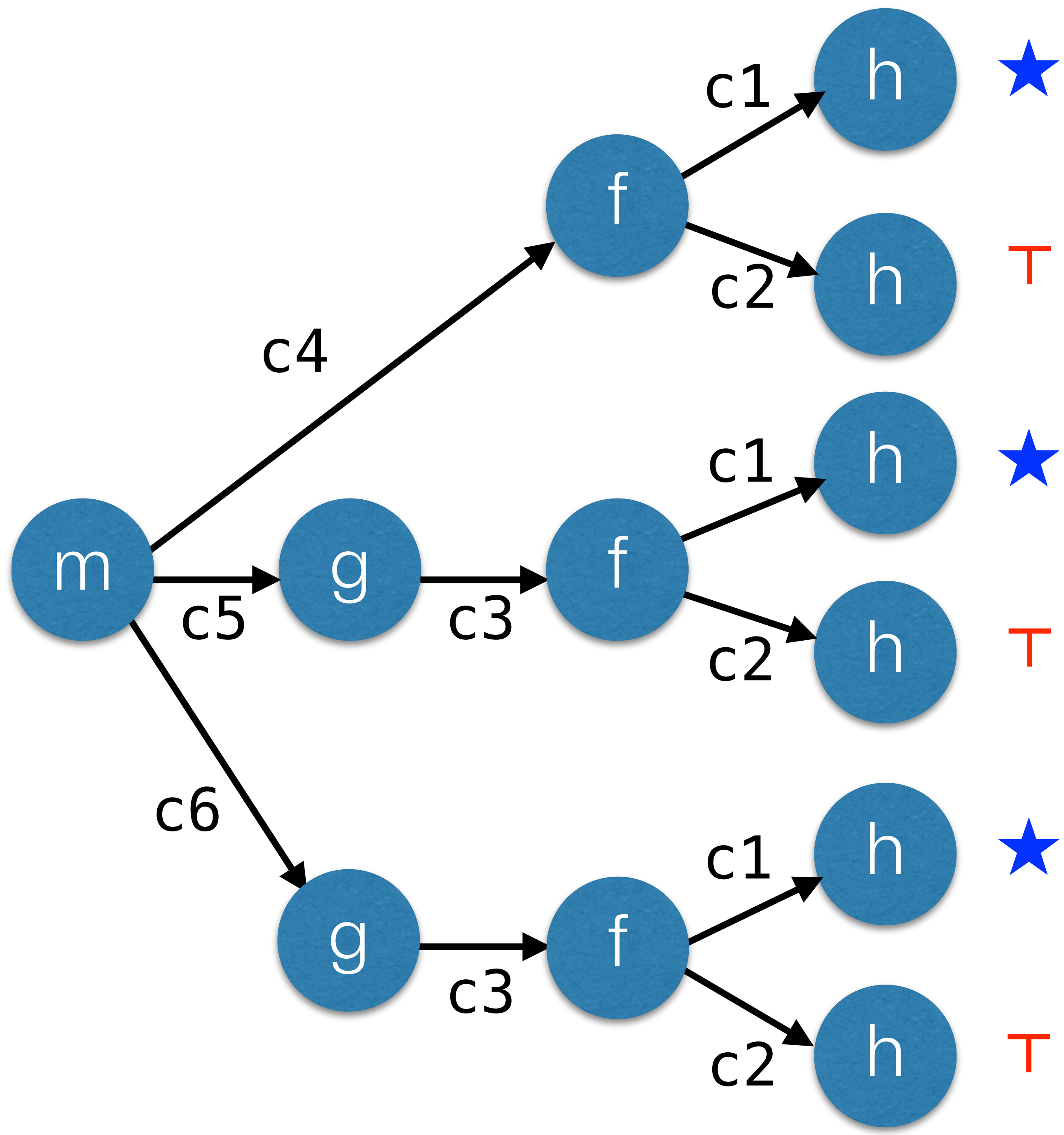
1. Collect queries whose expressions are assigned with ★

```
    int h(n) {ret n;}

    void f(a) {
c1: ★ x = h(a);
      assert(x > 1);  // Q1
c2: ⊤ y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



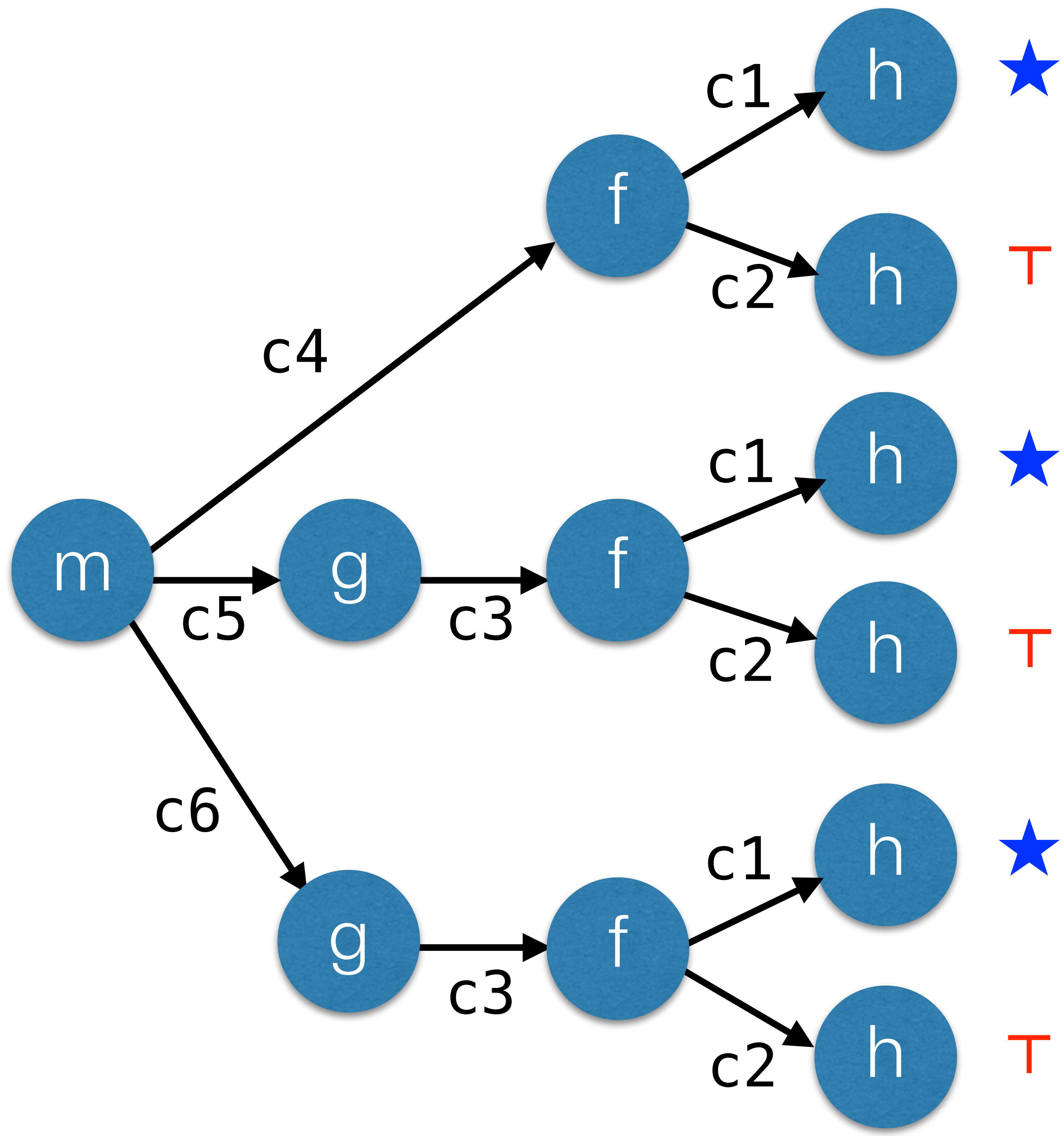
2. Find the program slice that contributes to the selected query

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



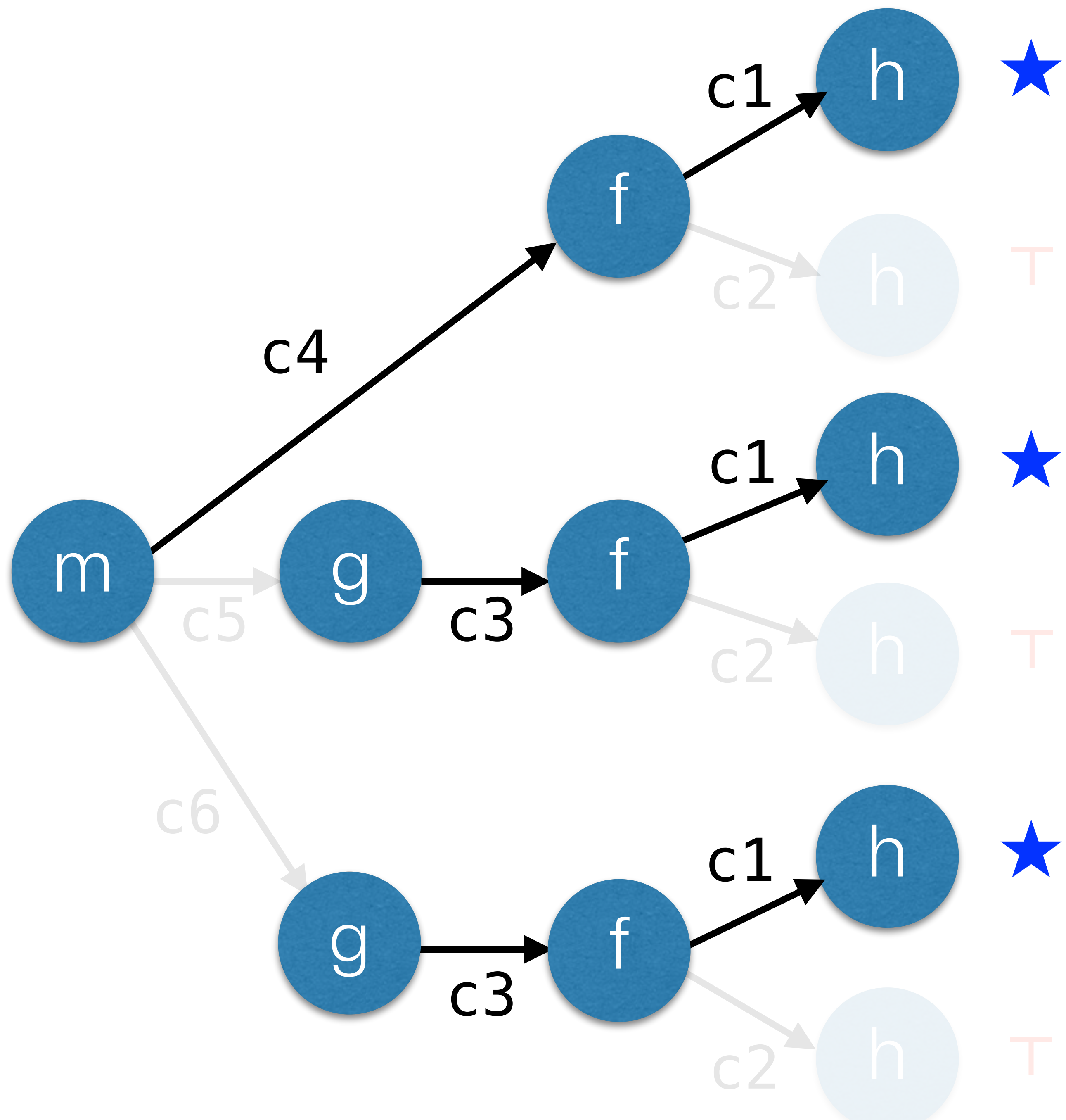
3. Collect contexts in the slice

```
    int h(n) {ret n;}

    void f(a) {
c1:   x = h(a);
      assert(x > 1);  // Q1
c2:   y = h(input());
      assert(y > 1);  // Q2
    }

c3: void g() {f(8);}

    void m() {
c4:   f(4);
c5:   g();
c6:   g();
    }
```



=> Contexts for **h**: {c3 · c1, c4 · c1}

Selective Relational Analysis

Relational Analysis

a = b

i < b

```
int a = b;
int c = input();
for (i = 0; i < b; i++) {
  assert (i < a);  // Q1
  assert (i < c);  // Q2
}
```

	a	b	c	i
a	0	0	∞	-1
b	0	0	∞	-1
c	∞	∞	0	∞
i	∞	∞	∞	0

$i - a \leq -1$

$i - c \leq \infty$

non-selective analysis

vs.

	a	b	i
a	0	0	-1
b	0	0	-1
i	∞	∞	0

our selective analysis

Impact Pre-Analysis

- Fully relational
- Approximated in other precision aspects

	a	b	c	i
a	0	0	∞	-1
b	0	0	∞	-1
c	∞	∞	0	∞
i	∞	∞	∞	0

octagon analysis

vs.

	a	b	c	i
a	★	★	⊤	★
b	★	★	⊤	★
c	⊤	⊤	★	⊤
i	⊤	⊤	⊤	★

impact pre-analysis

Experiments

- Implemented on top of Sparrow The Early Bird
    - Selective context-sensitive analysis
    - Selective octagon analysis
- Evaluated on 10 GNU benchmarks (2~100KLoC)

Selective Context-Sensitivity

		Context-Insensitve		Ours	
Pgm	**LOC**	**#alarms**	**time(s)**	**#alarms**	**time(s)**
spell	2K	58	0.6	30	0.9
bc	13K	606	14.0	483	16.2
tar	20K	940	42.1	799	47.2
less	23K	654	123.0	562	166.4
sed	27K	1,325	107.5	1,238	117.6
make	27K	1,500	88.4	1,028	106.2
grep	32K	735	12.1	653	15.9
wget	35K	1,307	69.0	942	82.1
a2ps	65K	3,682	118.1	2,121	177.7
bison	102K	1,894	136.3	1,742	173.4
TOTAL	**346K**	**12,701**	**707.1**	**9,598**	**903.6**

24.4%

Selective Context-Sensitivity

		Context-Insensitve		Ours	
Pgm	LOC	#alarms	time(s)	#alarms	time(s)
spell	2K	58	0.6	30	0.9
bc	13K	606	14.0	483	16.2
tar	20K	940	42.1	799	47.2
less	23K	654	123.0	562	166.4
sed	27K	1,325	107.5	1,238	117.6
make	27K	1,500	88.4	1,028	106.2
grep	32K	735	12.1	653	15.9
wget	35K	1,307	69.0	942	82.1
a2ps	65K	3,682	118.1	2,121	177.7
bison	102K	1,894	136.3	1,742	173.4
TOTAL	346K	12,701	707.1	9,598	903.6

pre-analysis : 14.7%
main analysis: 13.1%

27.8%

k-CFA did not scale

- 2 or 3-CFA did not scale over 10KLoC
  - e.g., for spell (2KLoC):
    - 3-CFA reported 30 alarms in 11.9s
    - cf) ours: 30 alarms in 0.9s
- 1-CFA did not scale over 40KLoC

Selective Octagon Analysis

			Existing Approach [Miné06]		Ours	
Pgm	LOC	#queries	proven	time(s)	proven	time(s)
calc	298	10	2	0.3	10	0.2
spell	2,213	16	1	4.8	16	2.4
barcode	4,460	37	16	11.8	37	30.5
httptunnel	6,174	28	16	26.0	26	15.3
bc	13,093	10	2	247.1	9	117.3
tar	20,258	17	7	1043.2	17	661.8
less	23,822	13	0	3031.5	13	2849.4
a2ps	64,590	11	0	29473.3	11	2741.7
TOTAL	135,008	142	44	33840.3	139	6418.6

+95

Selective Octagon Analysis

			Existing Approach [Miné06]		Ours	
Pgm	**LOC**	**#queries**	**proven**	**time(s)**	**proven**	**time(s)**
calc	298	10	2	0.3	10	0.2
spell	2,213	16	1	4.8	16	2.4
barcode	4,460	37	16	11.8	37	30.5
httptunnel	6,174	28	16	26.0	26	15.3
bc	13,093	10	2	247.1	9	117.3
tar	20,258	17	7	1043.2	17	661.8
less	23,822	13	0	3031.5	13	2849.4
a2ps	64,590	11	0	29473.3	11	2741.7
TOTAL	**135,008**	**142**	**44**	**33840.3**	**139**	**6418.6**

reduce time by -81%

Summary

- A method for **precise** yet **scalable** static analysis
  - Impact pre-analysis + Selective main analysis
- Generally applicable
  - context-sensitivity, relational analysis, etc