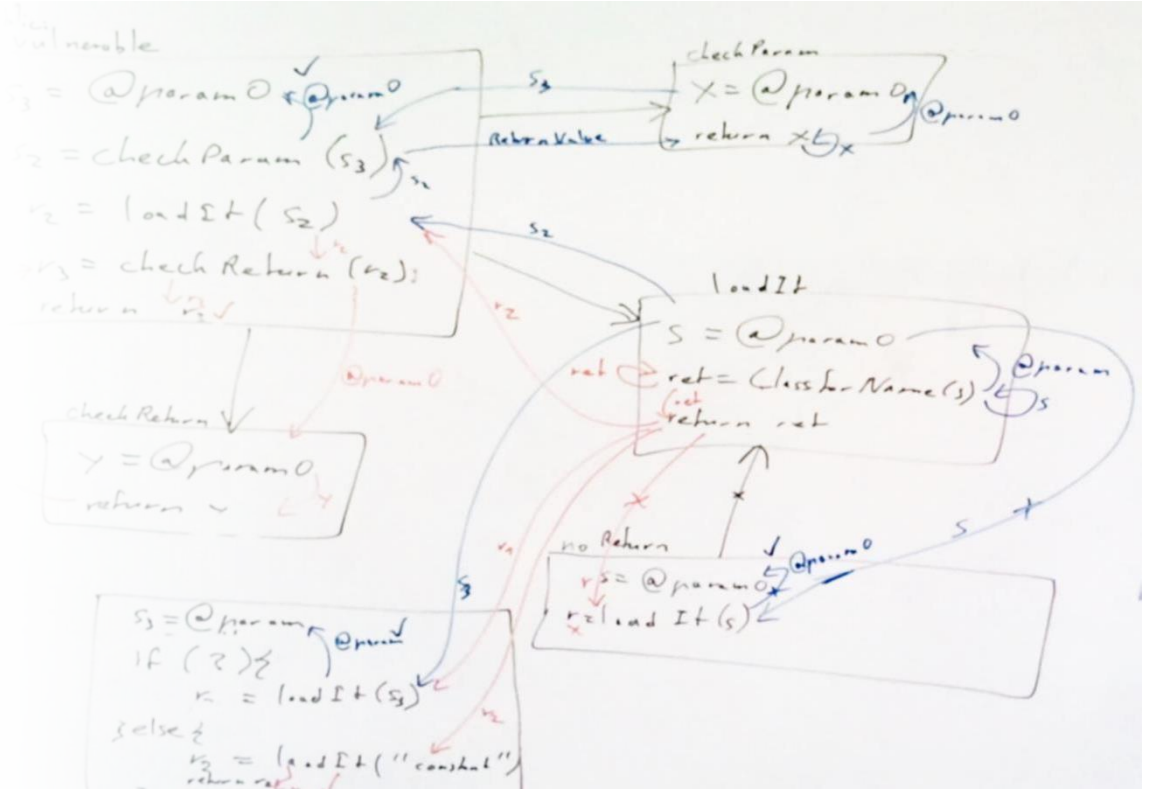
Access-Path Abstraction: Scaling Field-Sensitive Data-Flow Analysis With Unbounded Access Paths

Johannes Lerch, Johannes Späth, Eric Bodden, and Mira Mezini

TECHNISCHE UNIVERSITÄT DARMSTADT

HEROS

https://github.com/Sable/heros



@stg_darmstadt

SOFTWARE TECHNOLOGY GROUP

Scalable Field-Sensitive Taint Analysis

- Based on IFDS-Framework [Reps et al. 1995]
  - Context Sensitive
  - Flow Sensitive
- Field Sensitive

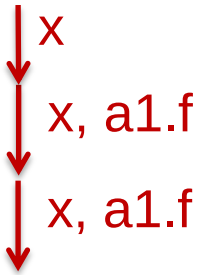
→ Scalability Issues

- Contributions of this work:
  - Identification of problematic cases
  - Approach solving these: IFDS-APA

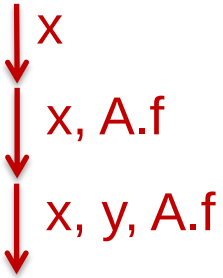
Tracking Fields

```
A a1 = new A()
A a2 = new A()
x = source()
a1.f = x
y = a2.f
sink(y)
```

Field Sensitive



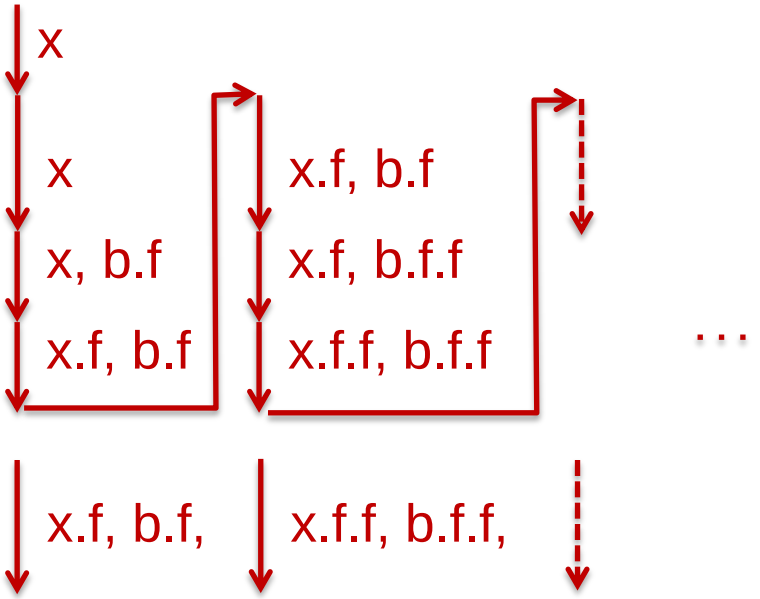
Field Based



False Warnings

Loops

```
x = source()
do {
    b = new A()
    b.f = x
    x = b
} while(…)
```



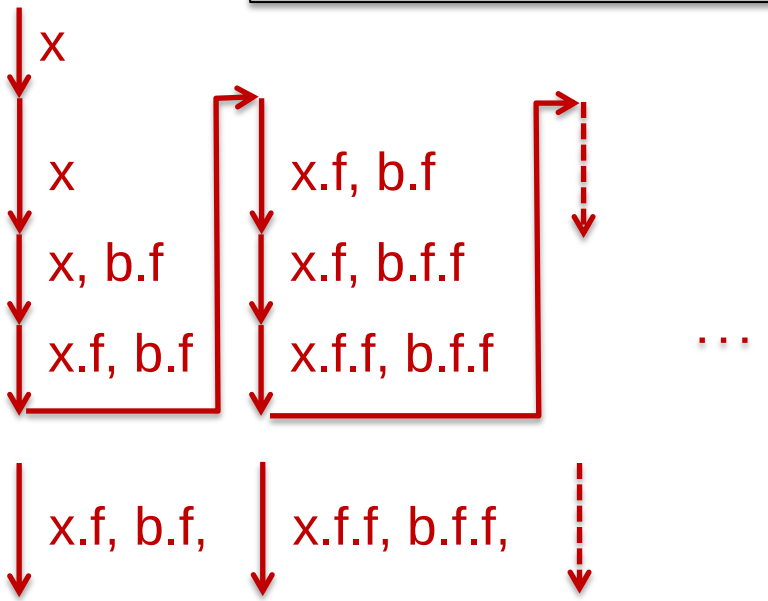
K-limiting

k-limiting [Jones et al. 1981]:

a.f.f.....f.f → a.f.f.....f.*

k k

```
x = source()
do {
    b = new A()
    b.f = x
    x = b
} while(…)
```



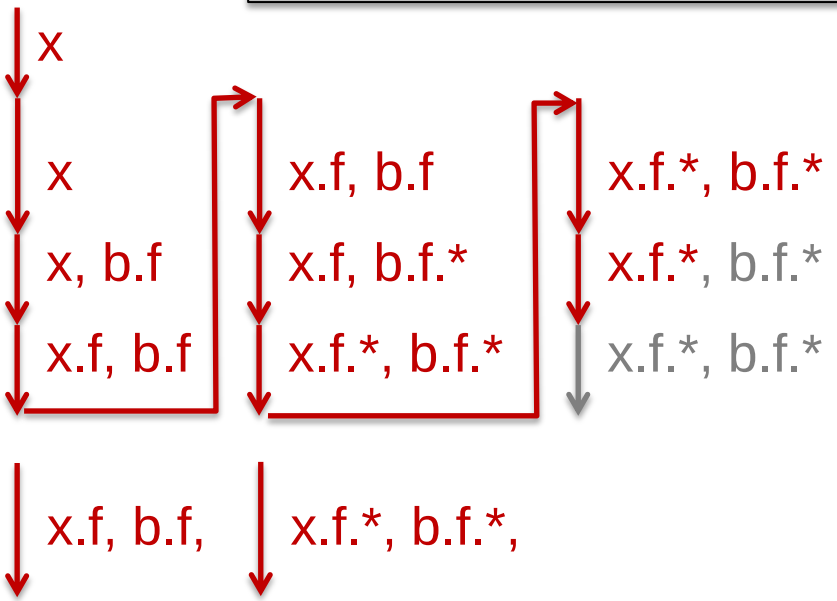
K-limiting

k-limiting [Jones et al. 1981]:

a.f.f.....f.f → a.f.f.....f.*

k k

```
x = source()
do {
    b = new A()
    b.f = x
    x = b
} while(…)
```

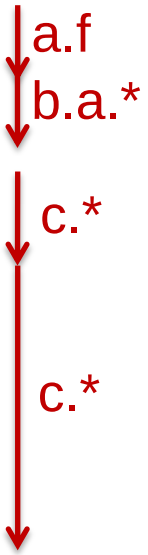


Large k → many facts

k-limiting with k = 1

Over-Approximation

```
foo() {
    a.f = source()
    b.a = a
    c = b.a.g
    bar(c)
}

bar(c) {
    sink(c)
}
```

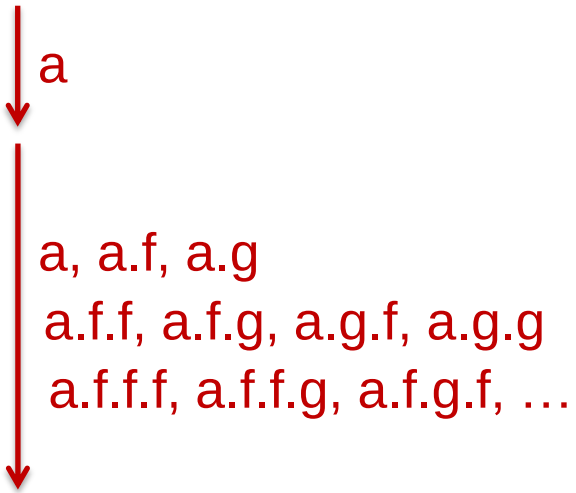


Small k → taints spread

k-limiting with k = 1

State Explosion

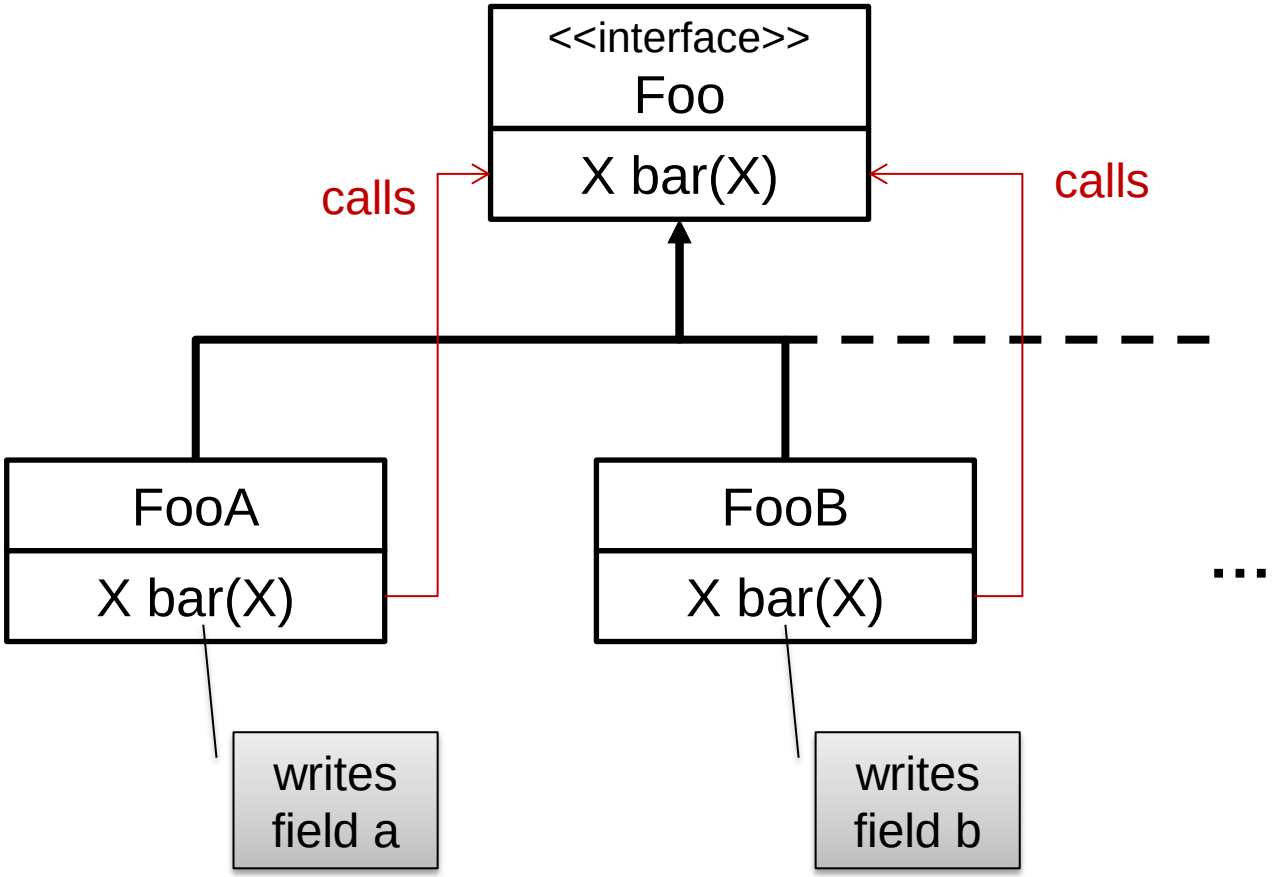
```
a = source()
while(…) {
  if(…)
      a.f = a
  else
      a.g = a
}
```



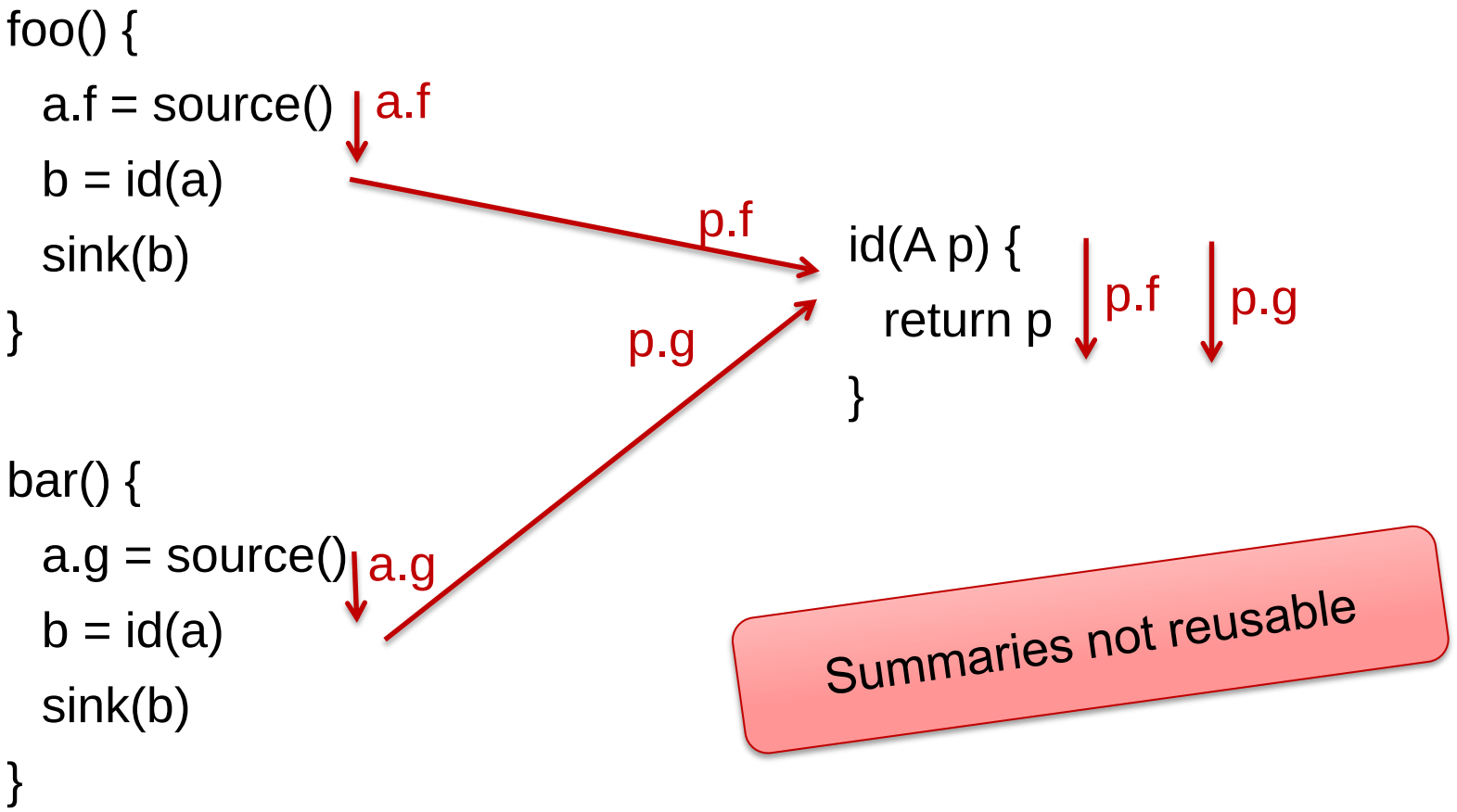
yields $\sum_{i=1}^{k} n^i$ different Access Paths for *n* fields and *k*-limiting

Amount of facts explodes

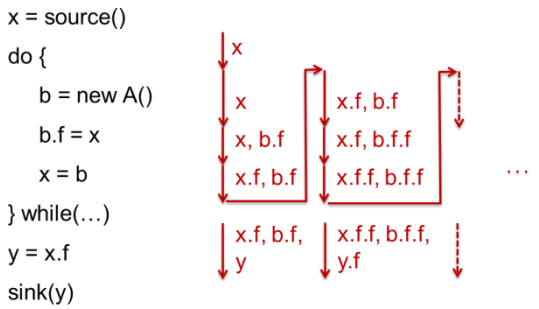
State Explosion



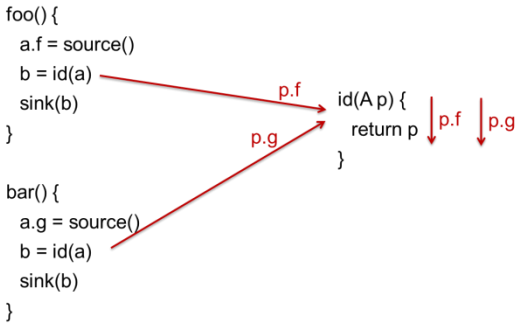
Summaries


foo() {
a.f = source()
b = id(a)
sink(b)
}
bar() {
a.g = source()
b = id(a)
sink(b)
}
a.f
p.f
id(A p) {
return p
}
p.f
p.g
p.g
a.g
Summaries not reusable

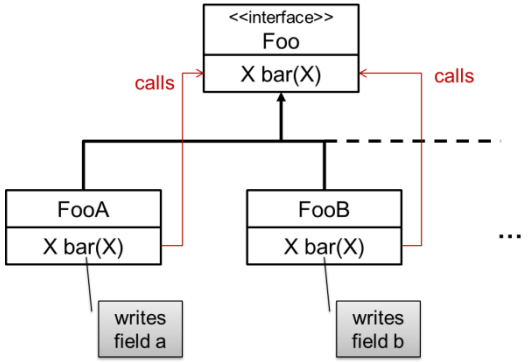
Identified Problems



Finite Domain

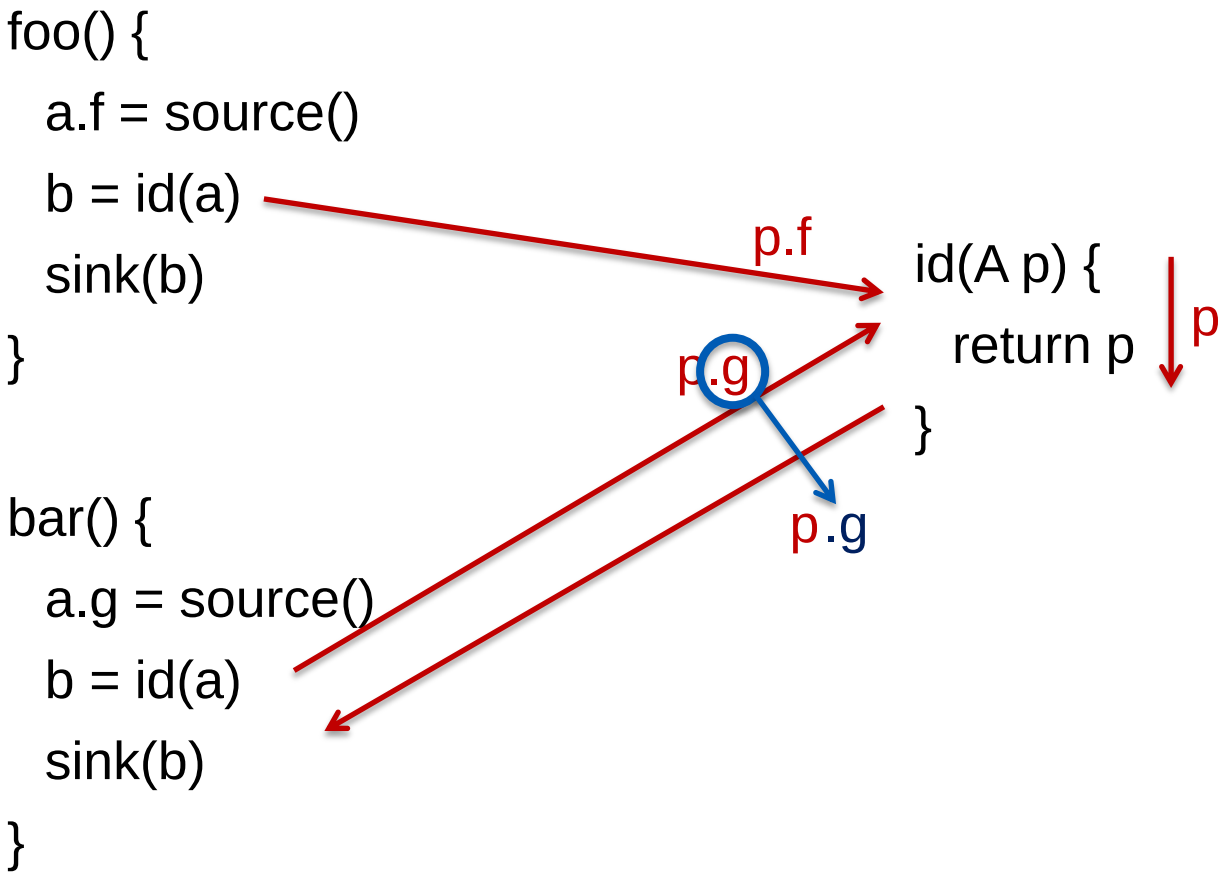


Reusability of Summaries

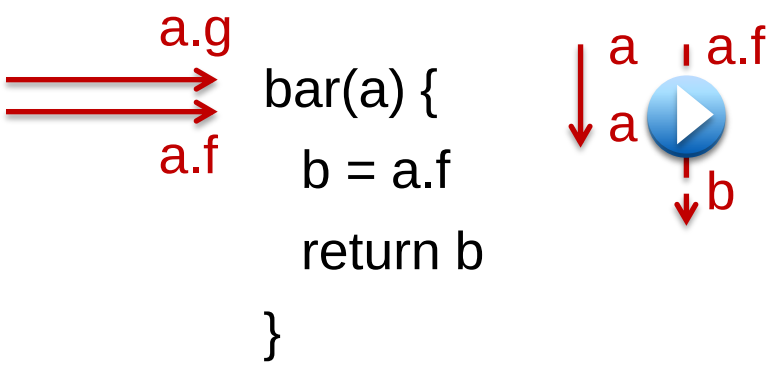


State Explosion

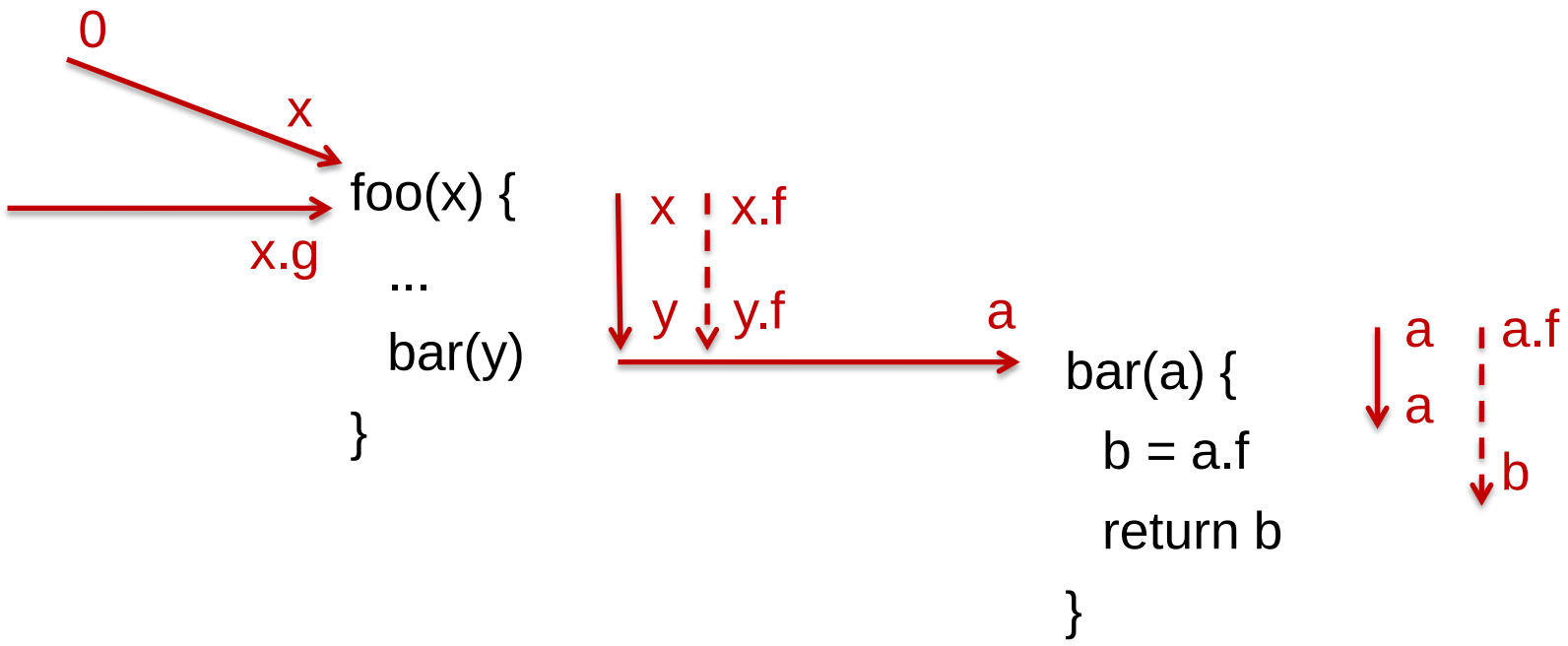
Abstract Summaries



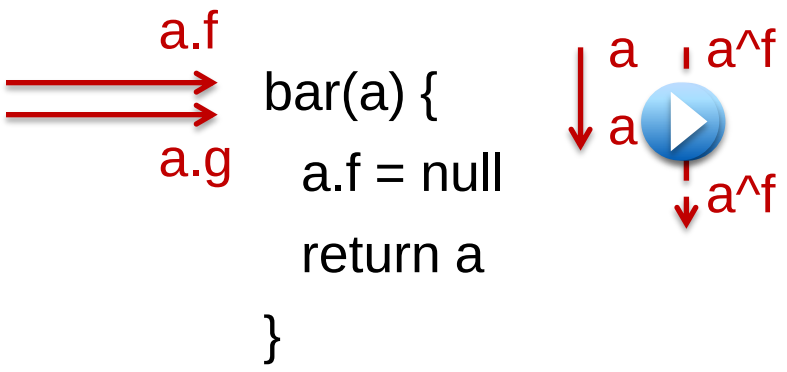
Field Read



Field Read – Transitive Check

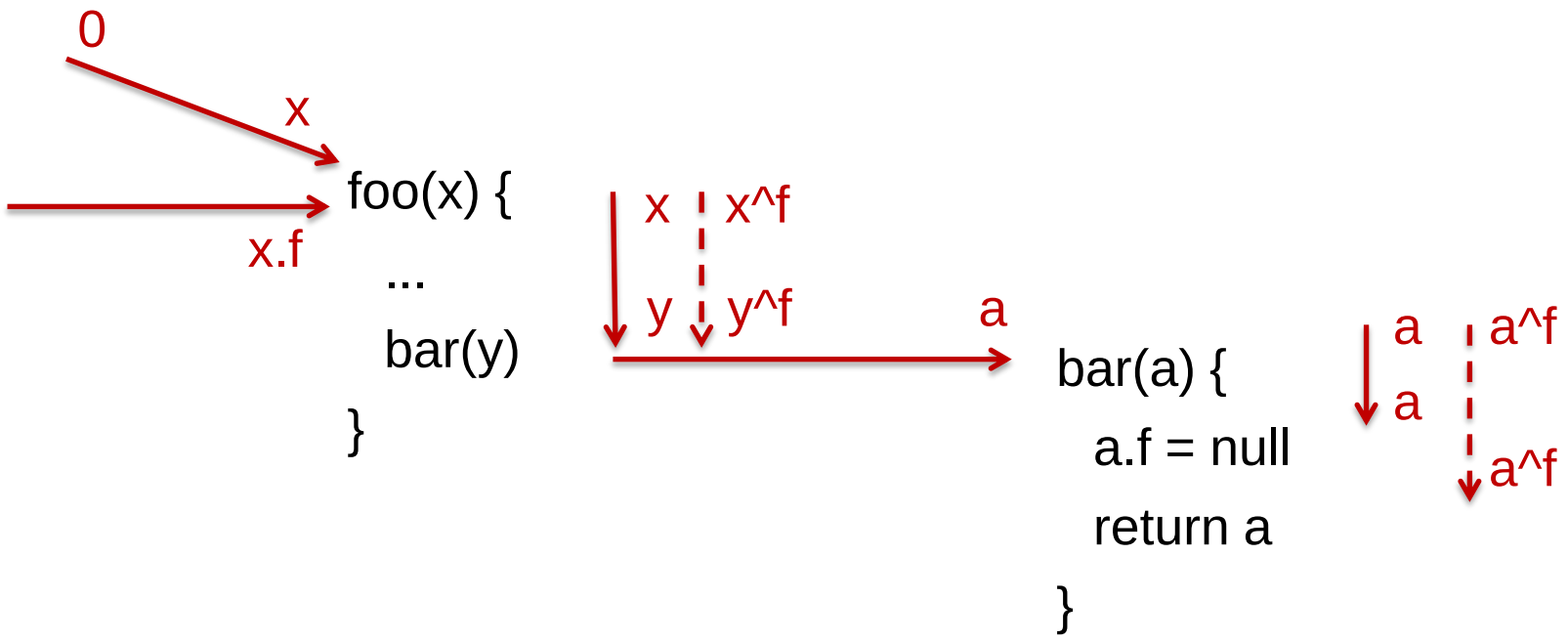


Field Write



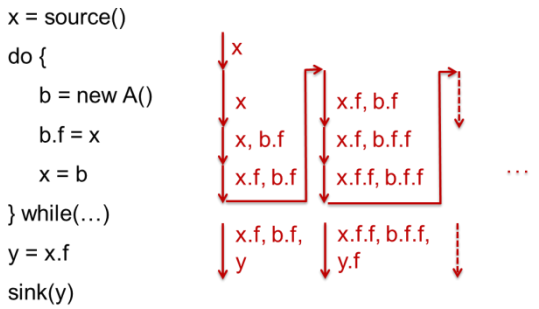
a^f short for a.*\{f}

Field Write – Transitive Check

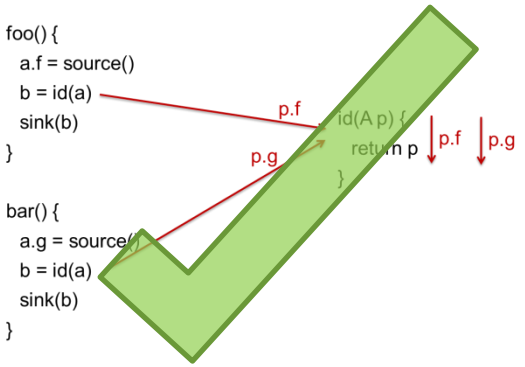


a^f short for a.*\{f}

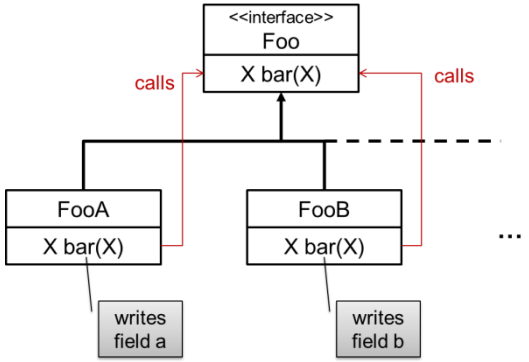
Identified Problems



Finite Domain



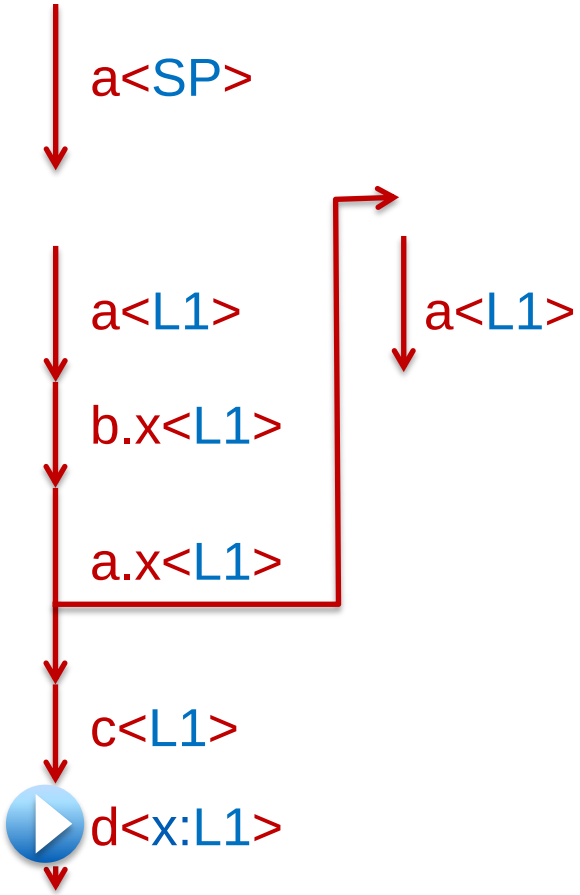
Reusability of Summaries



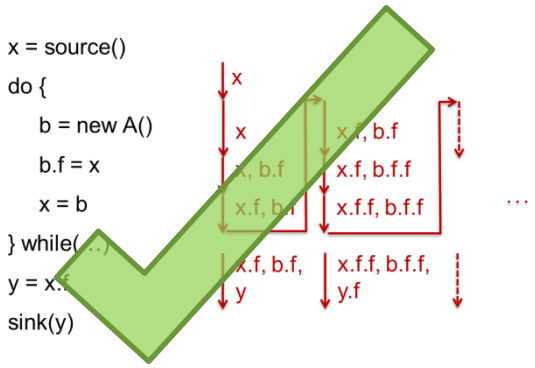
State Explosion

Abstraction Points

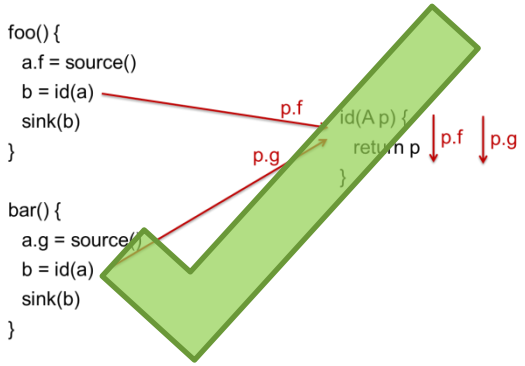
```
SP: foo(a) {

L1:     while(…) {
            b = new A()
            b.x = a
            a = b
        }
        c = a.x
        d = c.x
```



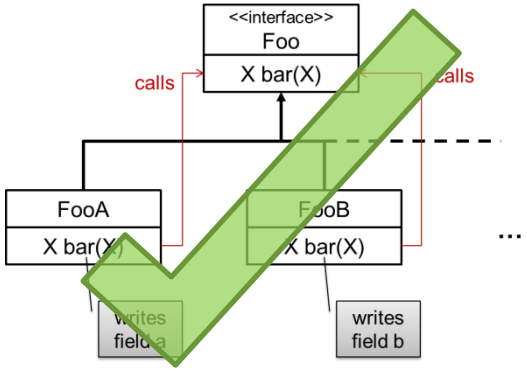
Identified Problems



Finite Domain



Reusability of Summaries



State Explosion

Evaluation

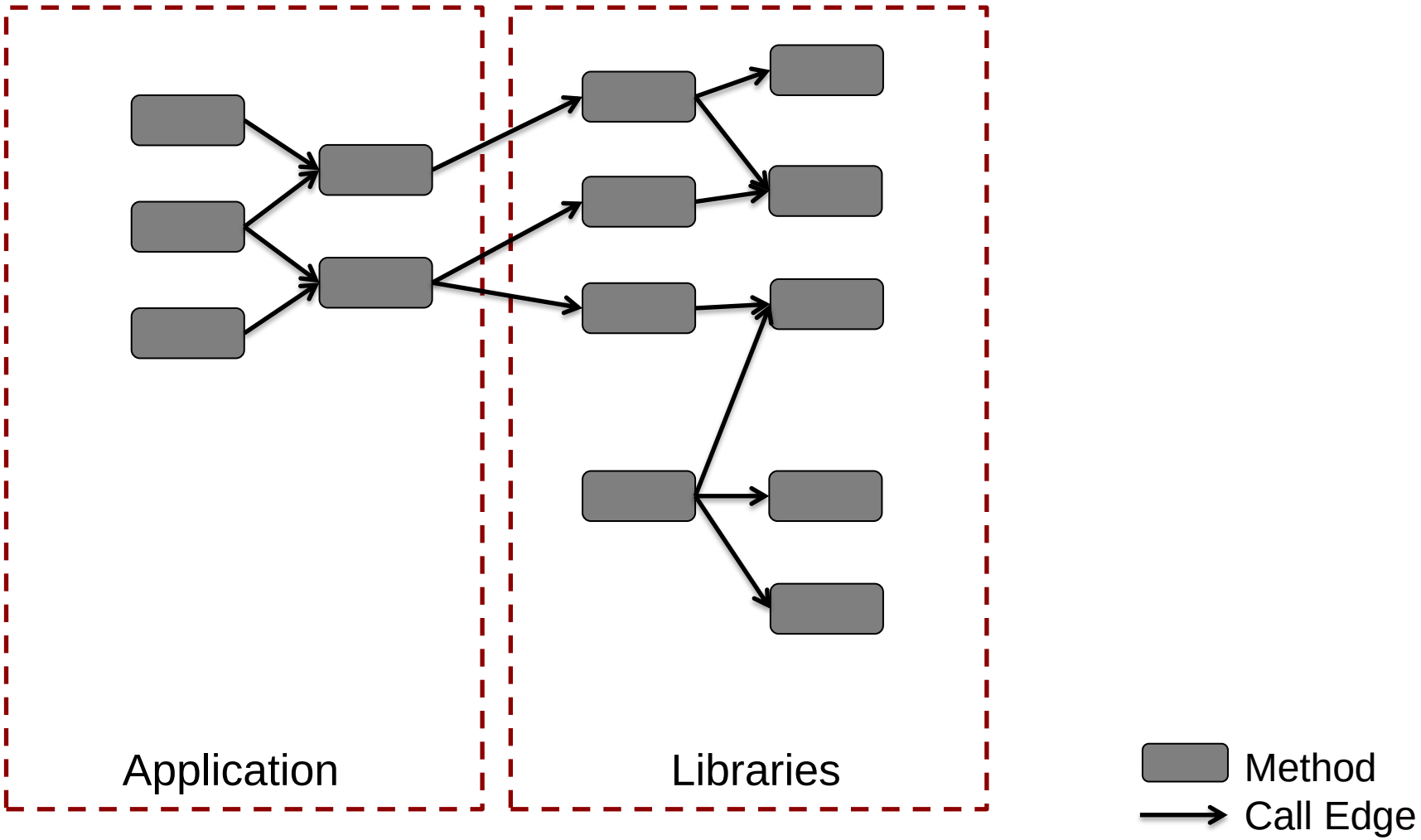
- SecuriBench
  - Benchmark consisting of 7 web applications
    - Including all their dependencies
    - especially the Java Class Library
  - Taint analysis for SQL injection, command injection, path traversal, unchecked redirection

Evaluation – SecuriBench Including Dependencies

Project	IFDS-APA	K-limiting				Field Based
		k=3	k=2	k=1	k=0	
blueblog	1.21	OoM				
jboard	322.70	OoM				
pebble	108.38	OoM				
personalblog	202.08	OoM				
roller	478.81	OoM				
snipsnap	307.65	OoM				
webgoat	57.75	OoM				

Run Time in Seconds

Visited Interprocedural Control-Flow Graph Edges

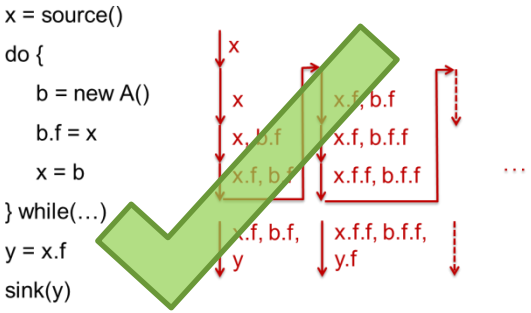


Evaluation – SecuriBench Including Dependencies

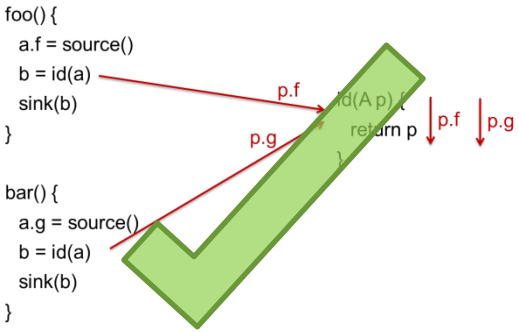
Project	ICFG Edges	IFDS-APA	K-limiting				Field Based
			k=3	k=2	k=1	k=0	
blueblog	692 483	3%	OoM	29%	29%	46%	8%
jboard	2 353 761	14%	OoM	OoM	OoM	61%	30%
pebble	1 769 459	13%	OoM	OoM	OoM	62%	27%
personalblog	2 194 345	15%	OoM	OoM	OoM	62%	28%
roller	2 891 553	15%	OoM	OoM	OoM	56%	27%
snipsnap	2 683 739	14%	OoM	OoM	OoM	63%	28%
webgoat	734 345	16%	OoM	44%	41%	52%	20%

Visited Interprocedural Control-Flow Graph Edges

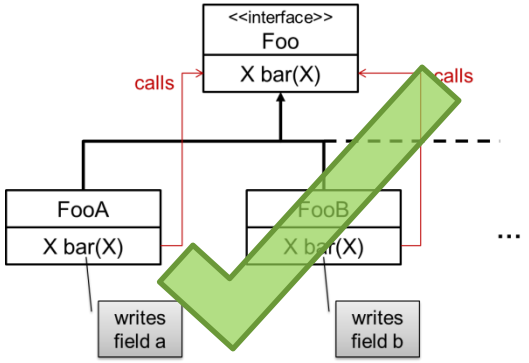
Summary



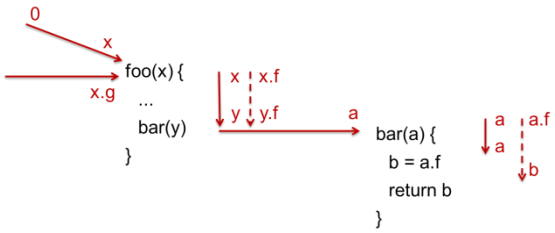
Finite Domain



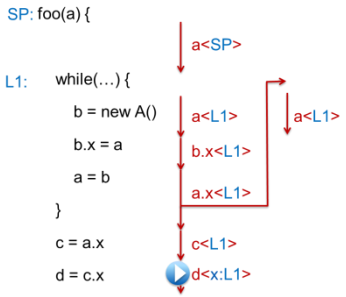
Reusability of Summaries



State Explosion



Caller Dependent Paused Edges



Abstraction Points

Project	IFDS-APA	K-limiting				Field Based
		k=3	k=2	k=1	k=0	
blueblog	1.21	OoM	54.56	43.54	27.15	1.05
jboard	322.70	OoM	OoM	OoM	228.84	40.81
pebble	108.38	OoM	OoM	OoM	138.40	17.13
personalblog	202.08	OoM	OoM	OoM	236.65	24.92
roller	478.81	OoM	OoM	OoM	102.83	35.19
snipsnap	307.65	OoM	OoM	OoM	203.16	113.01
webgoat	57.75	OoM	253.56	98.14	30.86	6.70

Scales as well as Field Based

Take away: *More precise* does not automatically mean *more expensive*