SHARP:
Fast Incremental Context-Sensitive Pointer Analysis for Java

Bozhen Liu, Jeff Huang

Texas A&M University, USA

TEXAS A&M UNIVERSITY

O2: Origin-Oriented Programming

Motivation

Whole Program

Eagle – *OOPSLA'19*
Context tunneling – *OOPSLA'18*
Scaler – *FSE'18*
Zipper – *OOPSLA'18*
Bean – *SAS'16*
Introspective analysis – *PLDI'14*
Client-driven pointer analysis – *SAS'03*
Refinement-based – *PLDI'06*
K-Obj – *TOSEM'05*
Cloning-based – *PLDI'04*

Access paths – *ECOOP'12*
VFG – *FSE'11*
AUX – *CGO'11*
SPAS – *APLAS'11*
Semi-sparse – *POPL'09*

Pointer Analysis

Context-sensitive
Field-sensitive
Flow-sensitive
…

Boomerang – *ECOOP'16*
CFL-reachability – *OOPSLA'05*
Demand-driven – *PLDI'01*

Incremental

IPA – *TOPLAS'19*
D4 – *PLDI'18*
Reviser – *ICSE'14*
Incremental CFL-reachability – *CC'13*
EMU – *ASE'12*
Query over logic program – *PPDP'05*
Incremental Landi-Ryder – *ICSE'99*

Motivation

Whole Program

Eagle – *OOPSLA'19*
Context tunneling – *OOPSLA'18*
Scaler – *FSE'18*
Zipper – *OOPSLA'18*
Bean – *SAS'16*
Introspective analysis – *PLDI'14*
Client-driven pointer analysis – *SAS'03*
Refinement-based – *PLDI'06*
K-Obj – *TOSEM'05*
Cloning-based – *PLDI'04*

Access paths – *ECOOP'12*
VFG – *FSE'11*
AUX – *CGO'11*
SPAS – *APLAS'11*
Semi-sparse – *POPL'09*

Boomerang – *ECOOP'16*
CFL-reachability – *OOPSLA'05*
Demand-driven – *PLDI'01*

Pointer Analysis

Context-sensitive
Field-sensitive
Flow-sensitive
...

Incremental

IPA – *TOPLAS'19*
D4 – *PLDI'18*
Reviser – *ICSE'14*
Incremental CFL-reachability – *CC'13*
EMU – *ASE'12*
Query over logic program – *PPDP'05*
Incremental Landi-Ryder – *ICSE'99*

Motivation

Whole Program

Eagle – *OOPSLA'19*
Context tunneling – *OOPSLA'18*
Scaler – *FSE'18*
Zipper – *OOPSLA'18*
Bean – *SAS'16*
Introspective analysis – *PLDI'14*
Client-driven pointer analysis – *SAS'03*
Refinement-based – *PLDI'06*
K-Obj – *TOSEM'05*
Cloning-based – *PLDI'04*

Access paths – *ECOOP'12*
VFG – *FSE'11*
AUX – *CGO'11*
SPAS – *APLAS'11*
Semi-sparse – *POPL'09*

Boomerang – *ECOOP'16*
CFL-reachability – *OOPSLA'05*
Demand-driven – *PLDI'01*

Pointer Analysis

Context-sensitive
Field-sensitive
Flow-sensitive
...

Incremental

IPA – *TOPLAS'19*
D4 – *PLDI'18*
Reviser – *ICSE'14*
Incremental CFL-reachability – *CC'13*
EMU – *ASE'12*
Query over logic program – *PPDP'05*
Incremental Landi-Ryder – *ICSE'99*

Motivation

Whole Program

Eagle – *OOPSLA'19*
Context tunneling – *OOPSLA'18*
Scaler – *FSE'18*
Zipper – *OOPSLA'18*
Bean – *SAS'16*
Introspective analysis – *PLDI'14*
Client-driven pointer analysis – *SAS'03*
Refinement-based – *PLDI'06*
K-Obj – *TOSEM'05*
Cloning-based – *PLDI'04*

Access paths – *ECOOP'12*
VFG – *FSE'11*
AUX – *CGO'11*
SPAS – *APLAS'11*
Semi-sparse – *POPL'09*

Pointer Analysis

Context-sensitive
Field-sensitive
Flow-sensitive
…

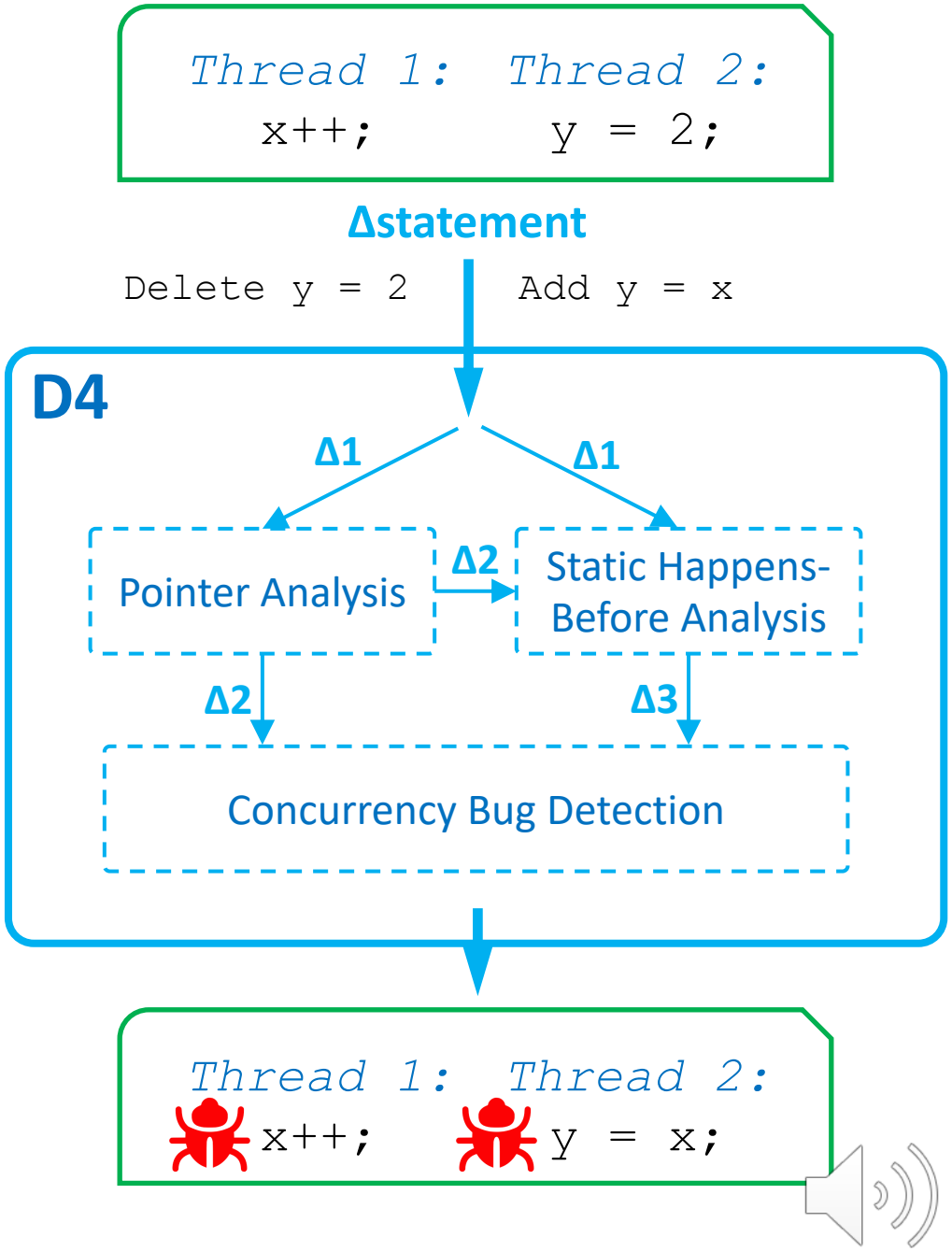
Boomerang – *ECOOP'16*
CFL-reachability – *OOPSLA'05*
Demand-driven – *PLDI'01*

Incremental

IPA – *TOPLAS'19*
D4 – *PLDI'18*
Reviser – *ICSE'14*
Incremental CFL-reachability – *CC'13*
EMU – *ASE'12*
Query over logic program – *PPDP'05*
Incremental Landi-Ryder – *ICSE'99*

IPA – *TOPLAS'19*
D4 – *PLDI'18*

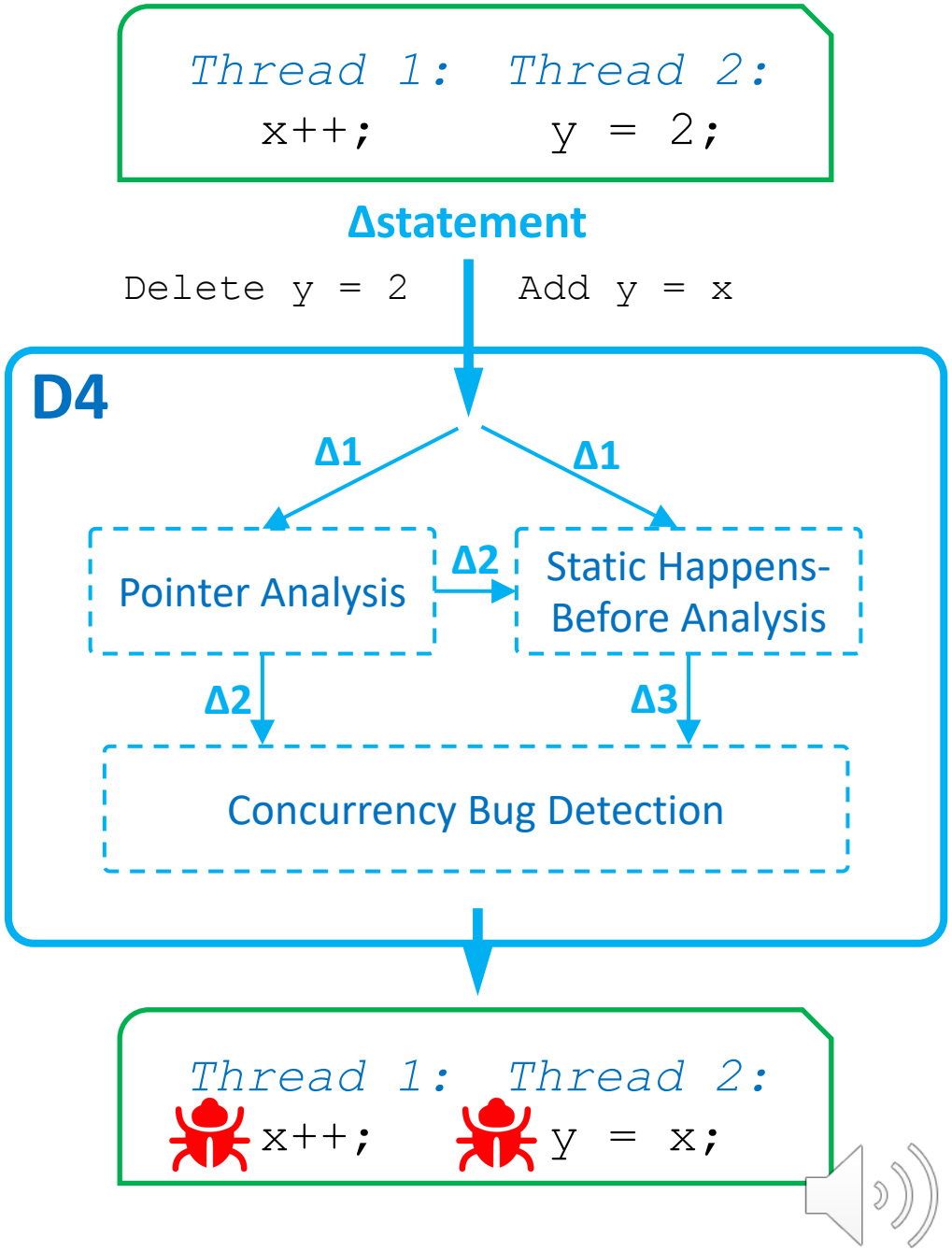
- **Key Insight**:
  - re-use result from previous analysis
  - analyze the code change ***only***
  - ***CG & PAG***
  - statement addition and ***deletion***
- ***Two steps*** for a deletion
  - Check & Propagate
- ***Parallel IPA (PIPA)***



IPA – *TOPLAS'19*
D4 – *PLDI'18*

- **Key Insight**:
  - re-use result from previous analysis
  - analyze the code change ***only***
  - ***CG & PAG***
  - statement addition and ***deletion***
- ***Two steps*** for a deletion
  - Check & Propagate
- ***Parallel IPA***

- **Context-Insensitive Pointer Analysis**
- Poor precision
- **~99%** False positives in Race Detection



#Reported Races by Using Different Whole Program Pointer Analysis Algorithm

App	No-Context	1-CFA	2-CFA	1-obj	2-obj
Avrora	12,633	45/99.6%	45/99.6%	47/99.6%	-
Batik	4,369	4,229/3.2%	640/85.4%	-	-
Eclipse	958	944/1.5%	822/14.2%	945/1.4%	-
H2	9,698	7,832/19.2%	6,322/34.8%	-	-
Jython	7,997	2,402/70.0%	2,358/70.5%	-	-
Luindex	3,218	2,821/12.3%	2,271/29.4%	-	-
Lusearch	567	538/5.1%	494/12.9%	529/6.7%	526/7.2%
Pmd	307	296/3.6%	293/4.6%	-	-
Sunflow	9,238	6,868/25.7%	5,899/36.1%	2,288/75.2%	-
Tomcat	751	701/6.7%	693/7.7%	585/22.1%	575/23.4%
Tradebeans	193	171/11.4%	168/13.0%	-	-
Tradesoap	264	179/32.2%	177/33.0%	-	-
Xalan	6	6/0.0%	6/0.0%	6/0.0%	-

- : time out after 4 hours.

Red: the percentage of reduced number of detected races compared with no-context.

Technical Challenges

- How to **incrementally** compute points-to results with **context-sensitivity**?
- How to guarantee its **soundness**?
  - statement deletion
  - method calls/object allocations
- How to **generalize** the incremental algorithm for different types of contexts,

  i.e., k-CFA and k-obj?

SHARP

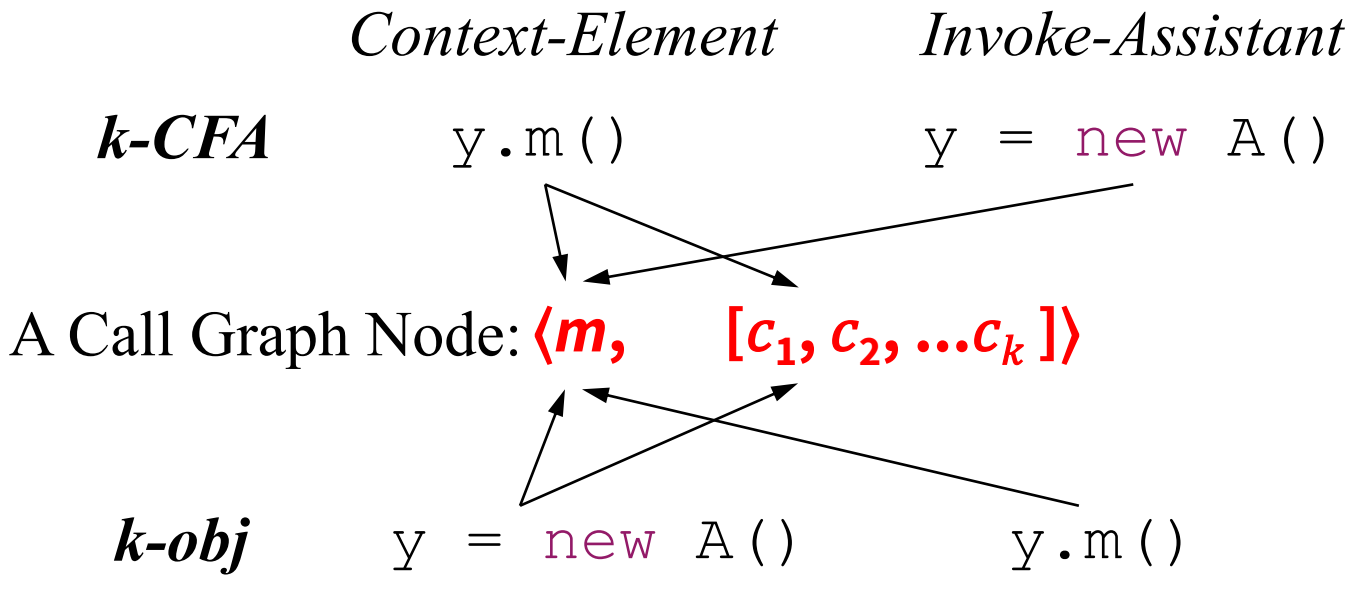
- Adapt ***IPA*** to both ***k-CFA*** and ***k-obj***
  - Identify ***redundant*** computation
  - ***Parallel*** for GitHub commit

SHARP

- Adapt ***IPA*** to both ***k-CFA*** and ***k-obj***
  - Identify ***redundant*** computation
  - ***Parallel*** for GitHub commit

```
public void m'() {
    y = new A();//O
    y.m();//S
    …
}

public void m() {
    p = new B();//O'
    …
}
```

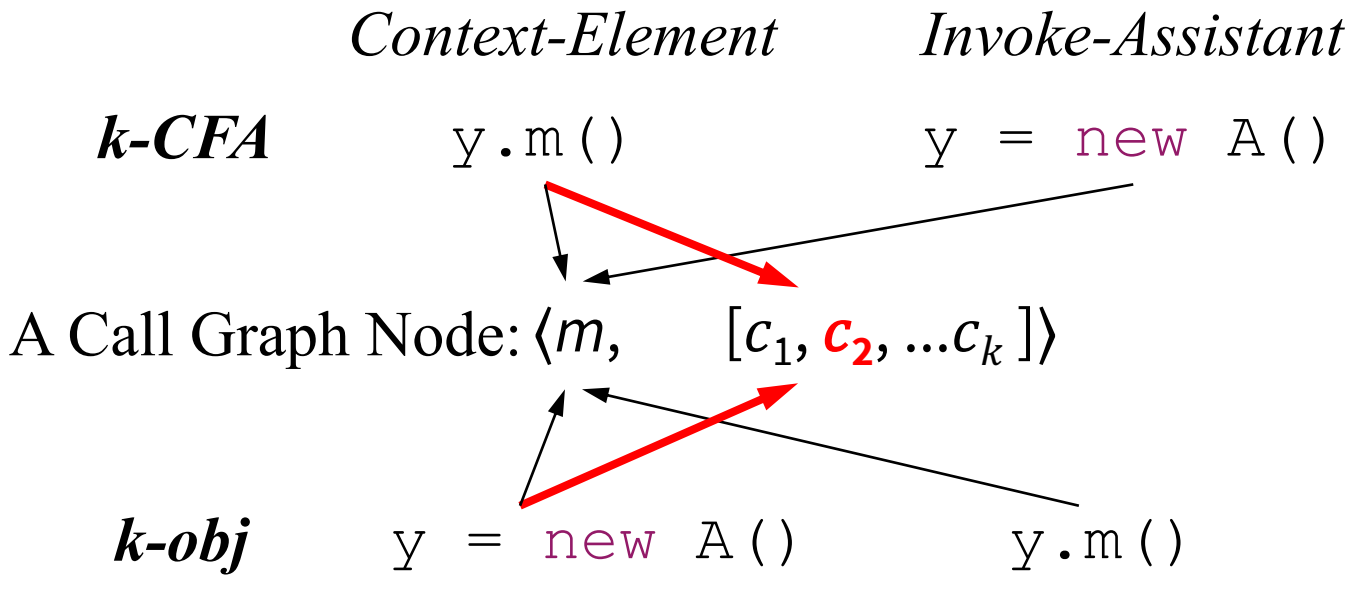


SHARP

- Adapt ***IPA*** to both ***k-CFA*** and ***k-obj***
  - Identify ***redundant*** computation
  - ***Parallel*** for GitHub commit

```
public void m'() {
    y = new A();//O
    y.m();//S
    …
}

public void m() {
    p = new B();//O'
    …
}
```

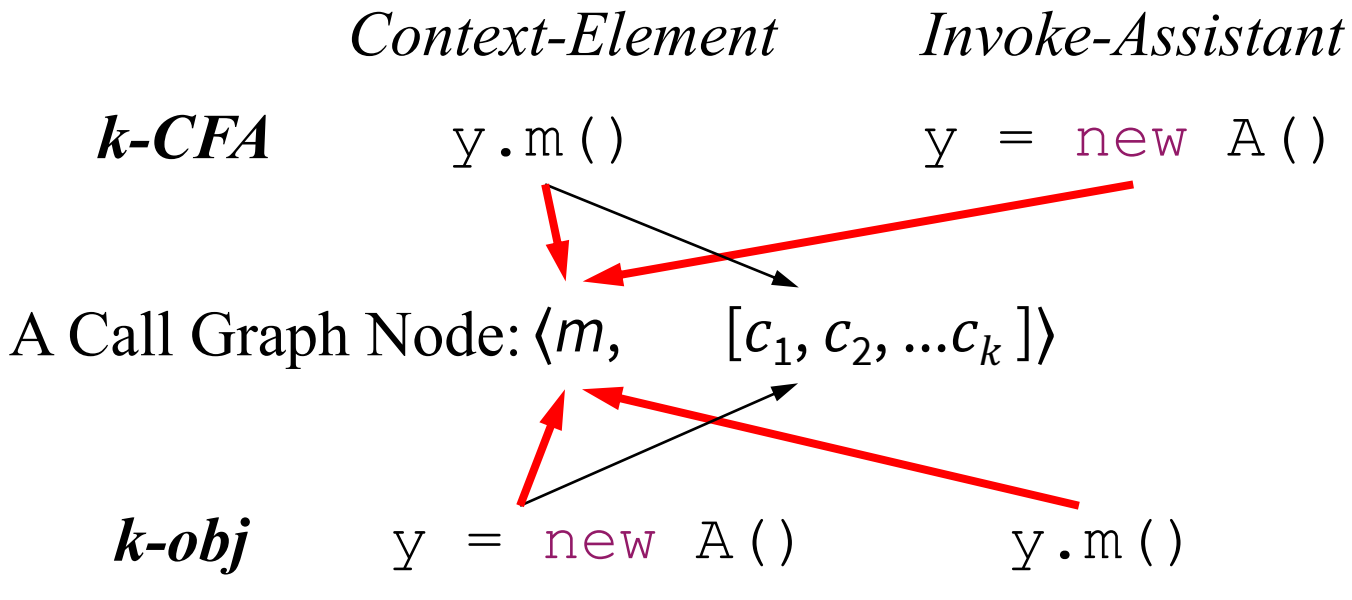


SHARP

- Adapt ***IPA*** to both ***k-CFA*** and ***k-obj***
  - Identify ***redundant*** computation
  - ***Parallel*** for GitHub commit

```
public void m'() {
    y = new A();//O
    y.m();//S
    …
}

public void m() {
    p = new B();//O'
    …
}
```

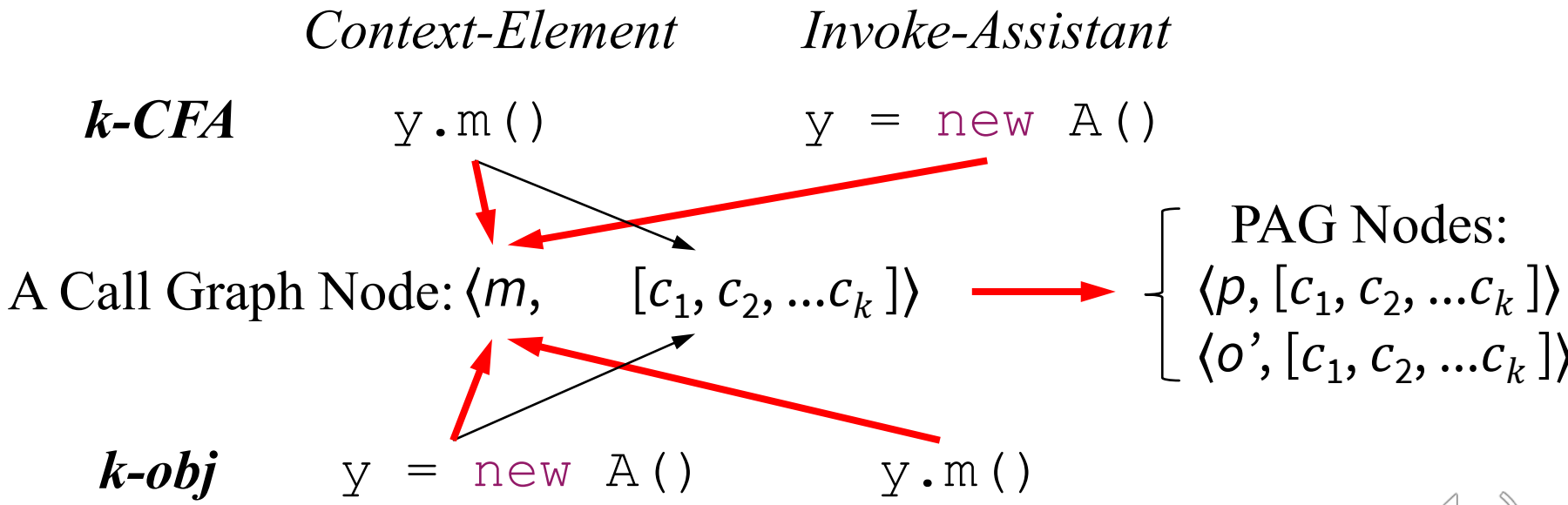


SHARP

- Adapt ***IPA*** to both ***k-CFA*** and ***k-obj***
  - Identify ***redundant*** computation
  - ***Parallel*** for GitHub commit

```
public void m'() {
    y = new A();//O
    y.m();//S
    …
}

public void m() {
    p = new B();//O'
    …
}
```



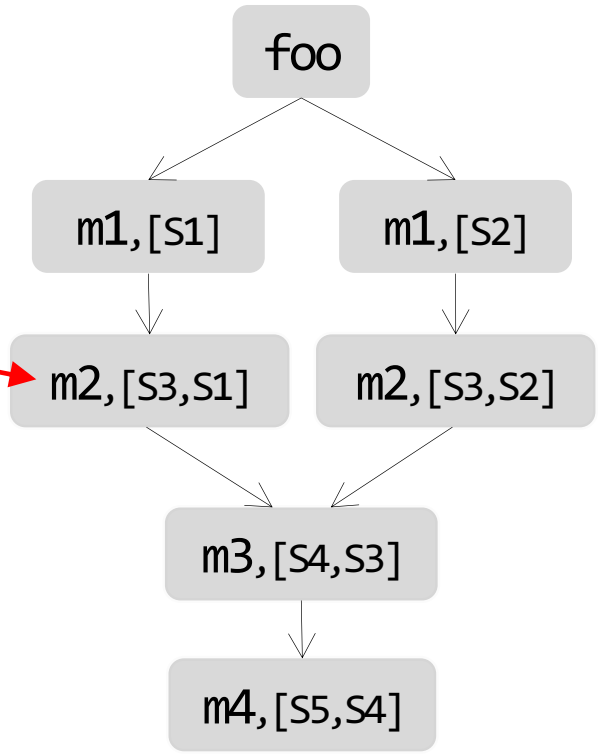
SHARP - Identify Redundant Computation

Code

```
1  public void foo() {
2    ...
3    a1.m1(b1);//S1
4    a2.m1(b2);//S2
5  }
6  public void m1(B p1){
7    m2(p1);//S3
8  }
9  public void m2(B p2) {
10   m3(p2);//S4  ...
11 }
12 public void m3(B p3) {
13   m4(p3);//S5  ...
14 }
15 public void m4(B p4) {
16   ...
17 }
```

Context [S3,S1]

CG

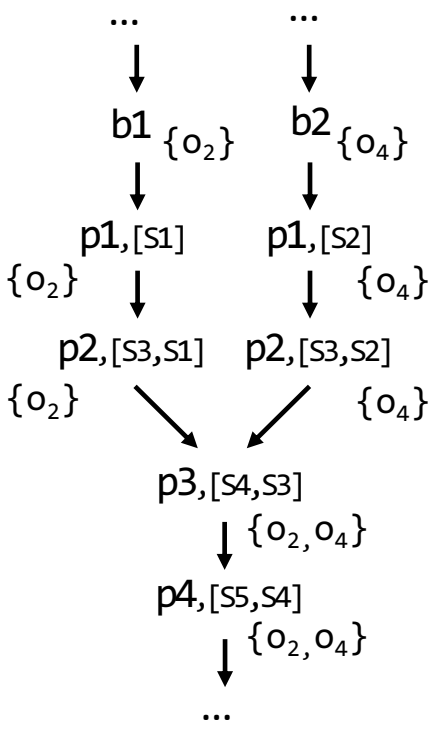


On-the-fly

2-CFA

(2-call-site)

PAG

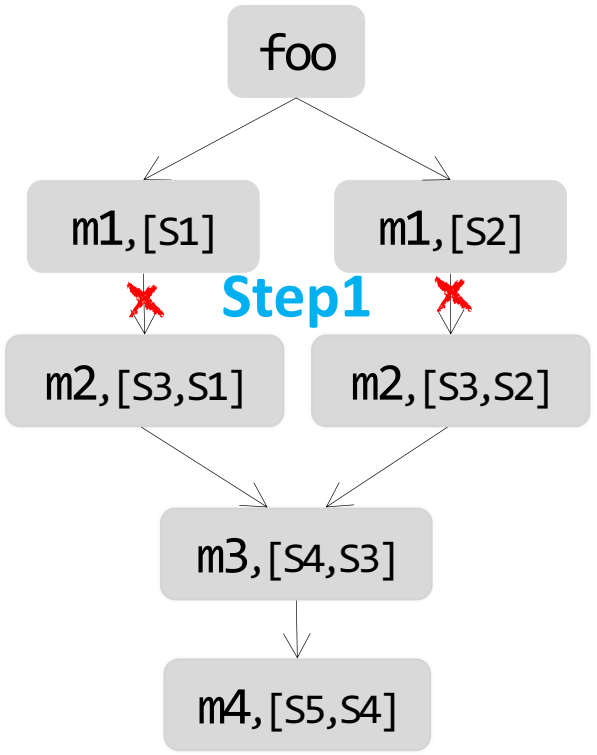


SHARP - Identify Redundant Computation

Code

```
1  public void foo() {
2     ...
3     a1.m1(b1);//S1
4     a2.m1(b2);//S2
5  }
6  public void m1(B p1){
7  -  m2(p1);//S3
8  }
9  public void m2(B p2) {
10    m3(p2);//S4   ...
11 }
12 public void m3(B p3) {
13    m4(p3);//S5   ...
14 }
15 public void m4(B p4) {
16    ...
17 }
```

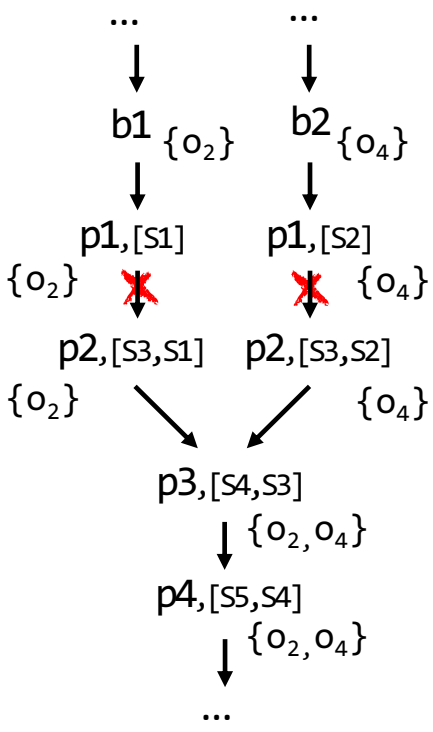
CG



On-the-fly

2-CFA

PAG

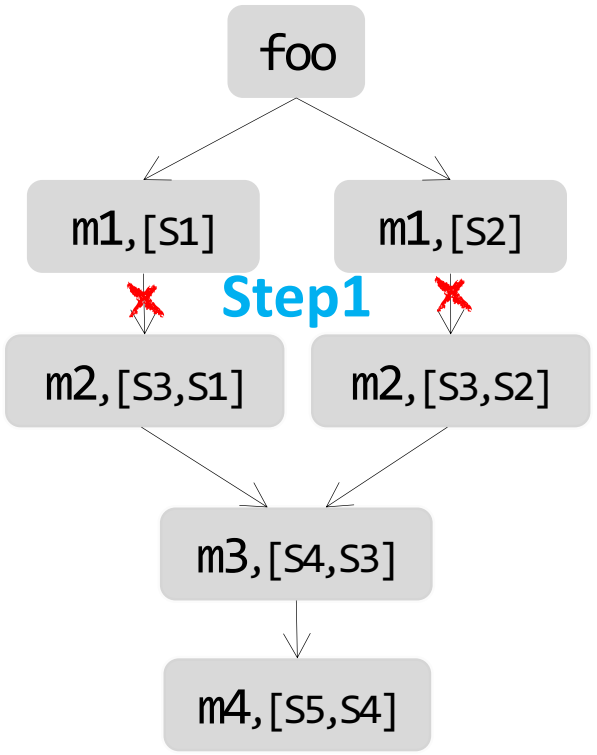


SHARP - Identify Redundant Computation

Code

```
1  public void foo() {
2     ...
3     a1.m1(b1);//S1
4     a2.m1(b2);//S2
5  }
6  public void m1(B p1){
7  -  m2(p1);//S3
8  }
9  public void m2(B p2) {
10    m3(p2);//S4   ...
11 }
12 public void m3(B p3) {
13    m4(p3);//S5   ...
14 }
15 public void m4(B p4) {
16    ...
17 }
```

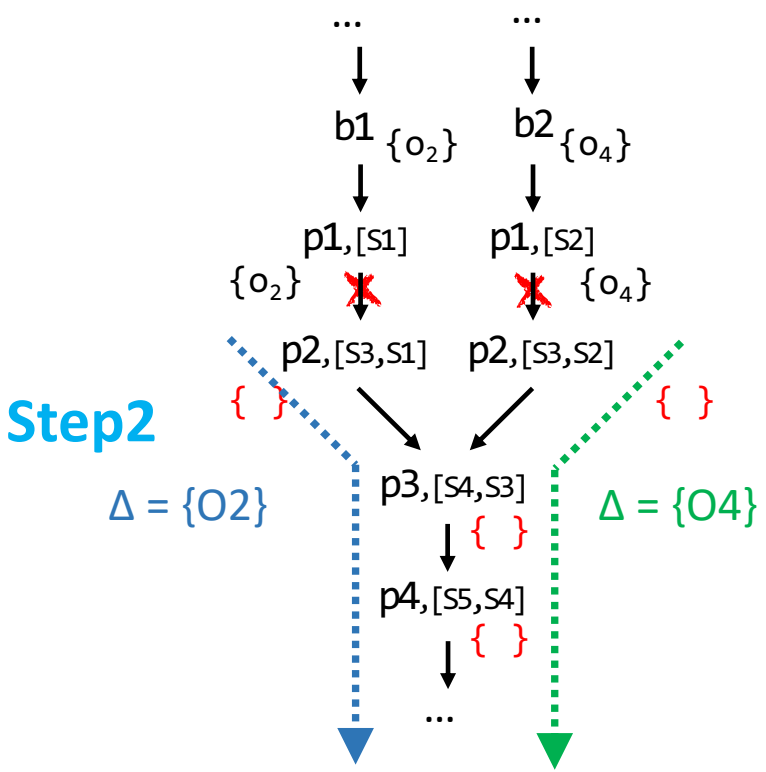
CG



On-the-fly

2-CFA

PAG

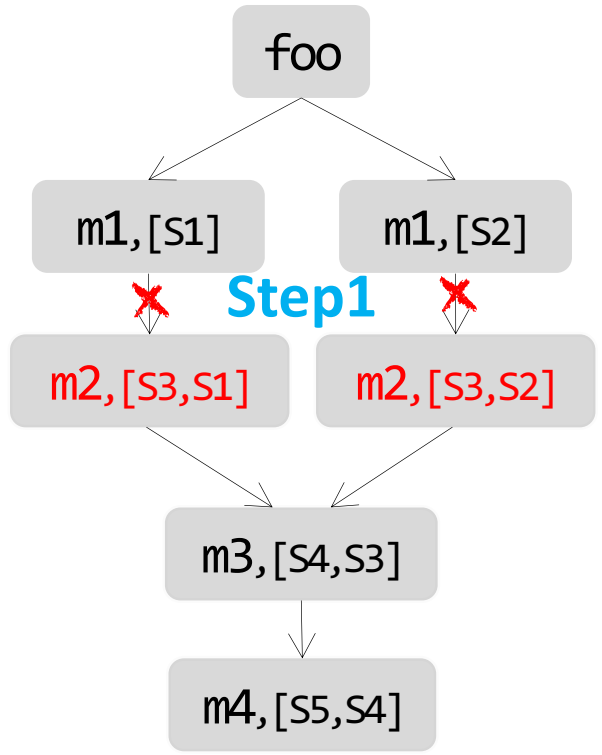


SHARP - Identify Redundant Computation

Code

```
1  public void foo() {
2     ...
3     a1.m1(b1);//S1
4     a2.m1(b2);//S2
5  }
6  public void m1(B p1){
7  -  m2(p1);//S3
8  }
9  public void m2(B p2) {
10    m3(p2);//S4  ...
11 }
12 public void m3(B p3) {
13    m4(p3);//S5   ...
14 }
15 public void m4(B p4) {
16    ...
17 }
```

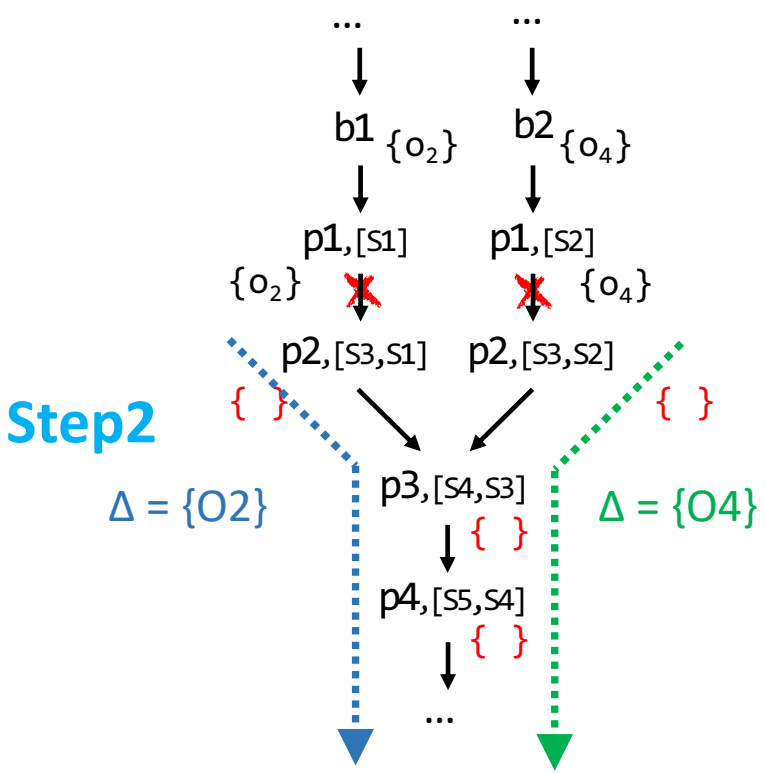
CG



On-the-fly

2-CFA

PAG

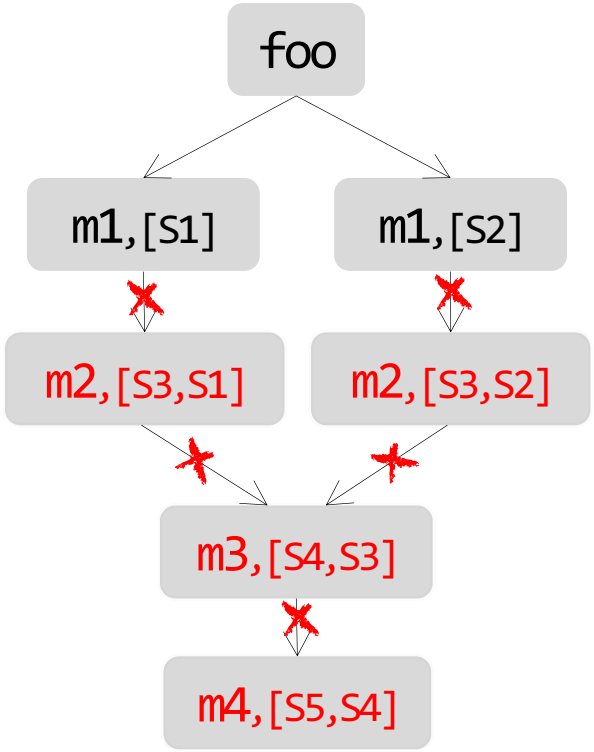


SHARP - Identify Redundant Computation

Code

```
1  public void foo() {
2     ...
3     a1.m1(b1);//S1
4     a2.m1(b2);//S2
5  }
6  public void m1(B p1){
7  -  m2(p1);//S3
8  }
9  public void m2(B p2) {
10    m3(p2);//S4   ...
11 }
12 public void m3(B p3) {
13    m4(p3);//S5   ...
14 }
15 public void m4(B p4) {
16    ...
17 }
```

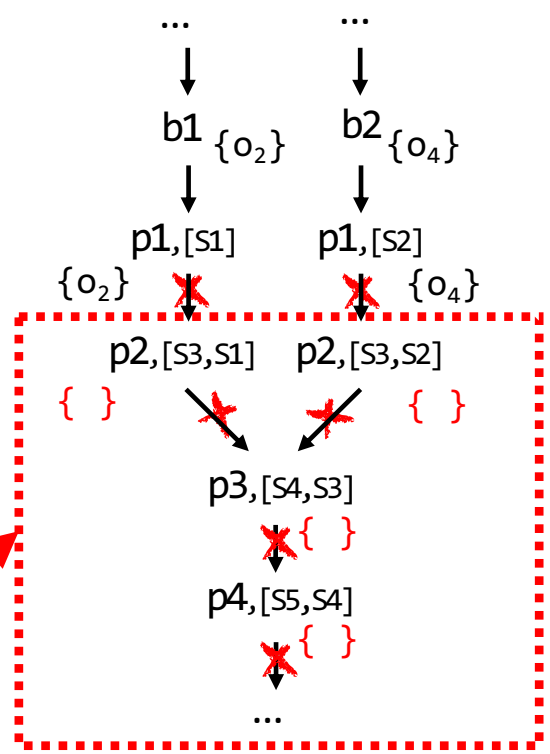
CG



On-the-fly

2-CFA

PAG



Redundant

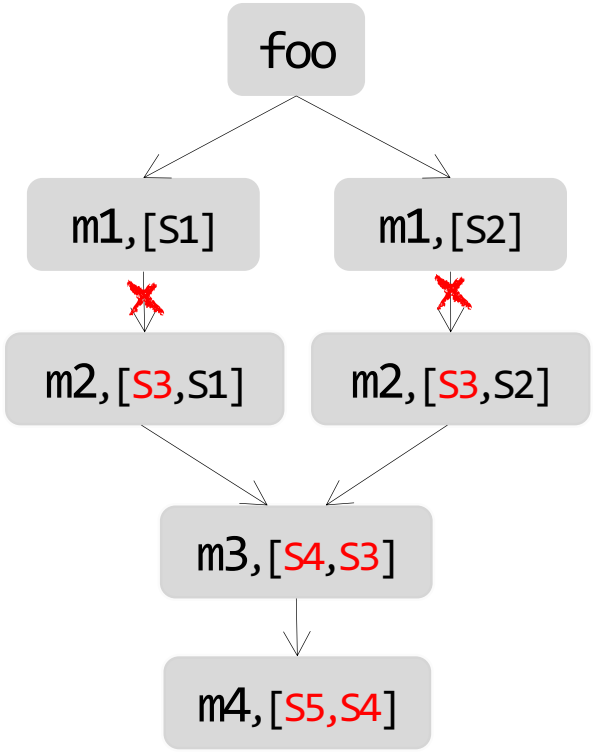
Check and Propagate

SHARP – Pre-Compute Invalid Context Elements

Code

```
1  public void foo() {
2     ...
3     a1.m1(b1);//S1
4     a2.m1(b2);//S2
5  }
6  public void m1(B p1){
7  -  m2(p1);//S3
8  }
9  public void m2(B p2) {
10    m3(p2);//S4   ...
11 }
12 public void m3(B p3) {
13    m4(p3);//S5   ...
14 }
15 public void m4(B p4) {
16    ...
17 }
```

CG



Invalid Elements

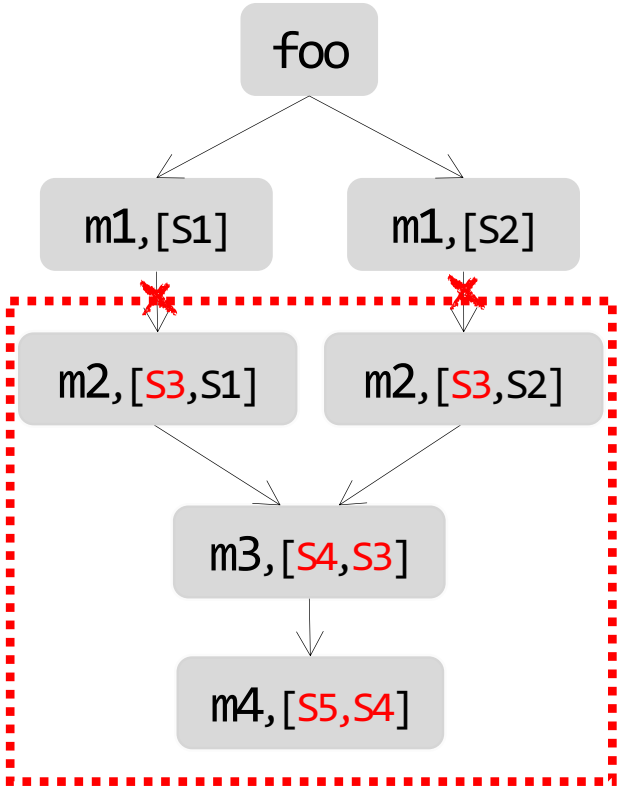
{ S3, S4, S5 }

SHARP – Pre-Compute Invalid Context Elements

Code

```
1  public void foo() {
2     ...
3     a1.m1(b1);//S1
4     a2.m1(b2);//S2
5  }
6  public void m1(B p1){
7  -  m2(p1);//S3
8  }
9  public void m2(B p2) {
10    m3(p2);//S4   ...
11 }
12 public void m3(B p3) {
13    m4(p3);//S5   ...
14 }
15 public void m4(B p4) {
16    ...
17 }
```

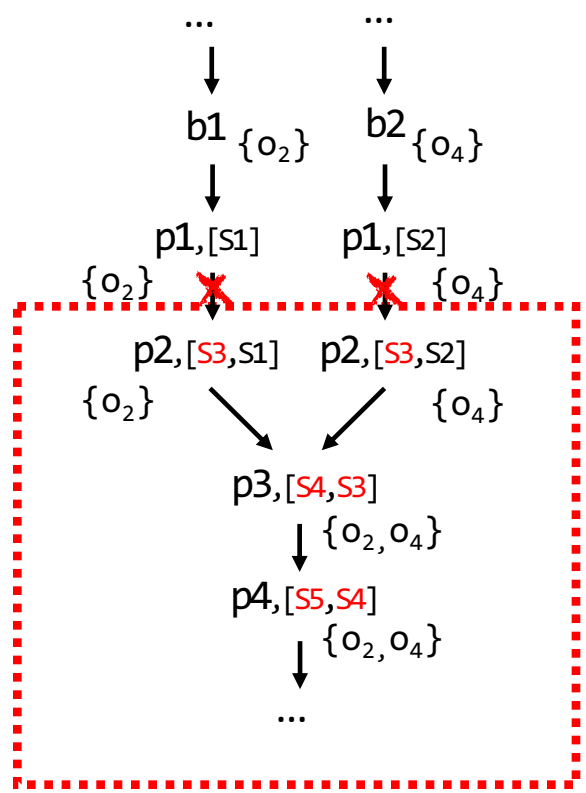
CG



Invalid Elements

{ S3, S4, S5 }

PAG

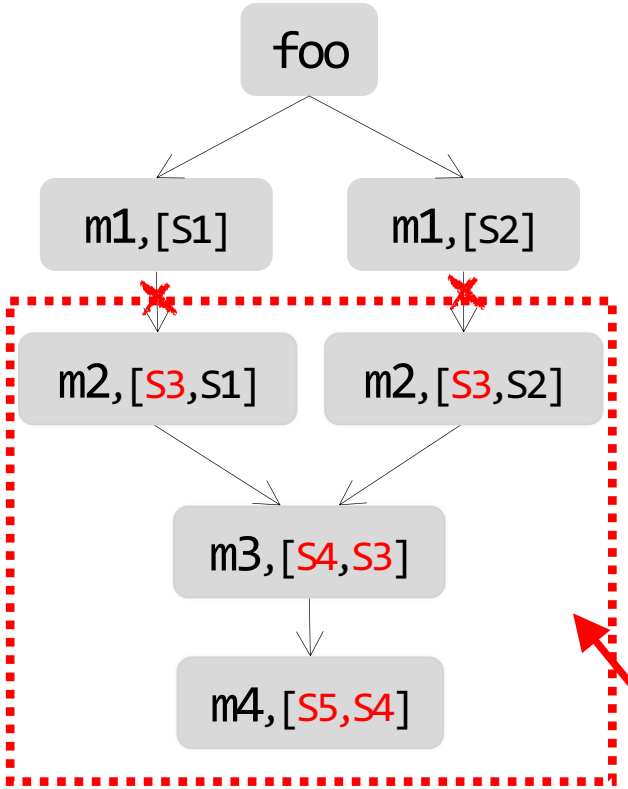


SHARP – Pre-Compute Invalid Context Elements

Code

```
1  public void foo() {
2     ...
3     a1.m1(b1);//S1
4     a2.m1(b2);//S2
5  }
6  public void m1(B p1){
7  -  m2(p1);//S3
8  }
9  public void m2(B p2) {
10    m3(p2);//S4   ...
11 }
12 public void m3(B p3) {
13    m4(p3);//S5   ...
14 }
15 public void m4(B p4) {
16    ...
17 }
```

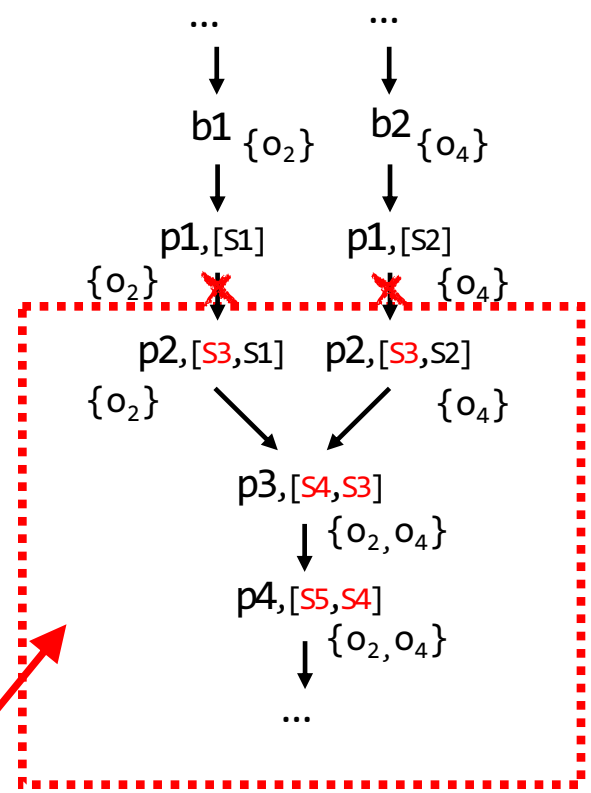
CG



Invalid Elements

{ S3, S4, S5 }

PAG



Delete

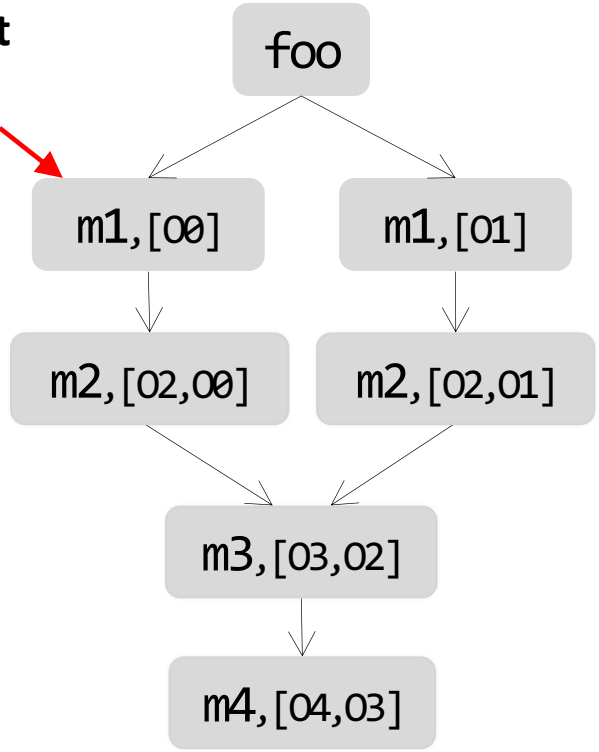
SHARP – Pre-Compute: 2-Obj

Code

```
1  public static void foo() {
2    x = new Obj();//O0
3    x.m1(b1);//pts(y)={O0}
4    y.m1(b2);//pts(y)={O1}
5  }
6  public void m1(B p1){
7    a.m2(p1);//pts(a)={O2}
8  }
9  public void m2(B p2) {
10   b.m3(p2);//pts(b)={O3}
11 }
12 public void m3(B p3) {
13   c.m4(p3);//pts(c)={O4}
14 }
15 public void m4(B p4) {
16   ...
17 }
```

Context [O0]

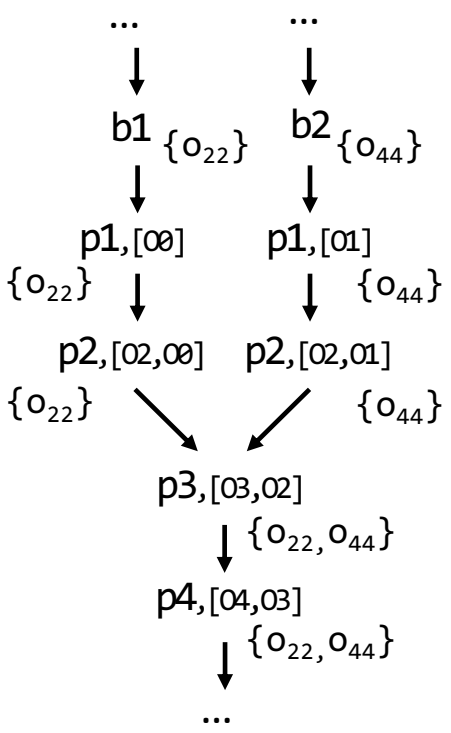
CG



On-the-fly

2-obj

PAG

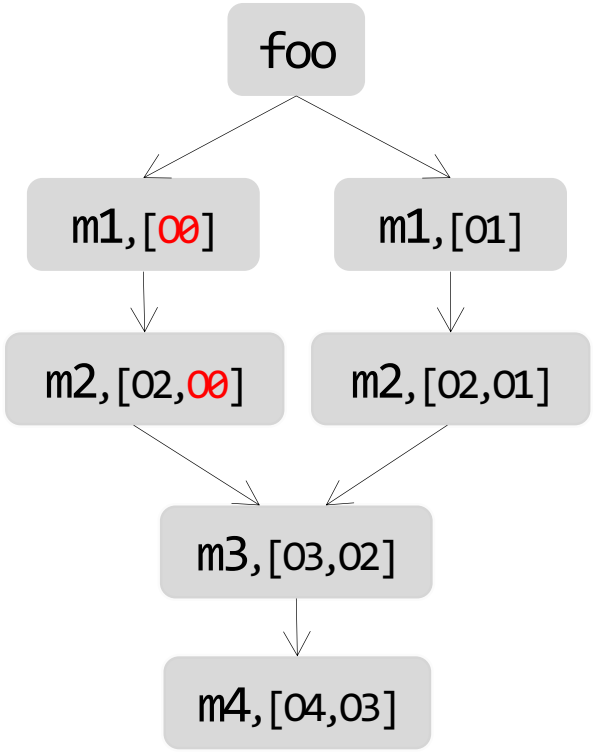


SHARP – Pre-Compute: 2-Obj

Code

```
1  public static void foo() {
2  -  x = new Obj();//O0
3     x.m1(b1);//pts(y)={O0}
4     y.m1(b2);//pts(y)={O1}
5  }
6  public void m1(B p1){
7     a.m2(p1);//pts(a)={O2}
8  }
9  public void m2(B p2) {
10    b.m3(p2);//pts(b)={O3}
11 }
12 public void m3(B p3) {
13    c.m4(p3);//pts(c)={O4}
14 }
15 public void m4(B p4) {
16    ...
17 }
```

CG



Invalid Elements

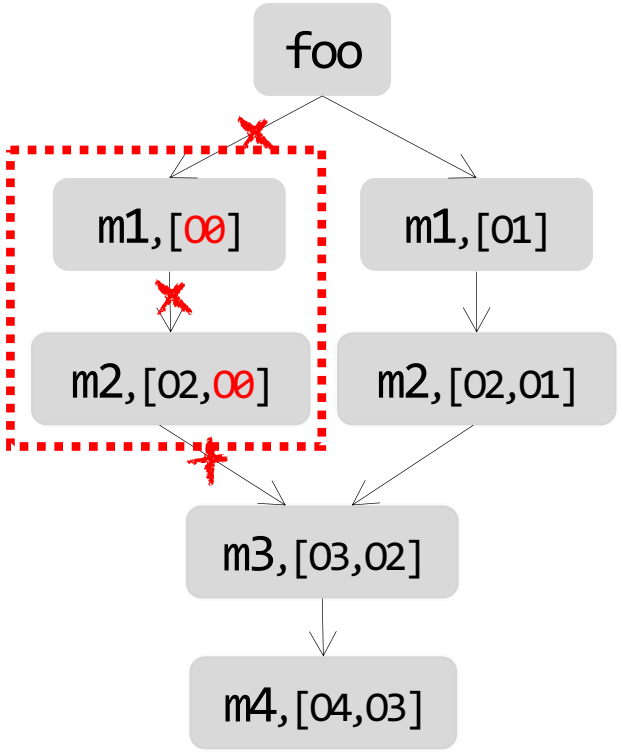
{ O0 }

SHARP – Pre-Compute: 2-Obj

Code

```
1  public static void foo() {
2  -  x = new Obj();//O0
3     x.m1(b1);//pts(y)={O0}
4     y.m1(b2);//pts(y)={O1}
5  }
6  public void m1(B p1){
7     a.m2(p1);//pts(a)={O2}
8  }
9  public void m2(B p2) {
10    b.m3(p2);//pts(b)={O3}
11 }
12 public void m3(B p3) {
13    c.m4(p3);//pts(c)={O4}
14 }
15 public void m4(B p4) {
16    ...
17 }
```

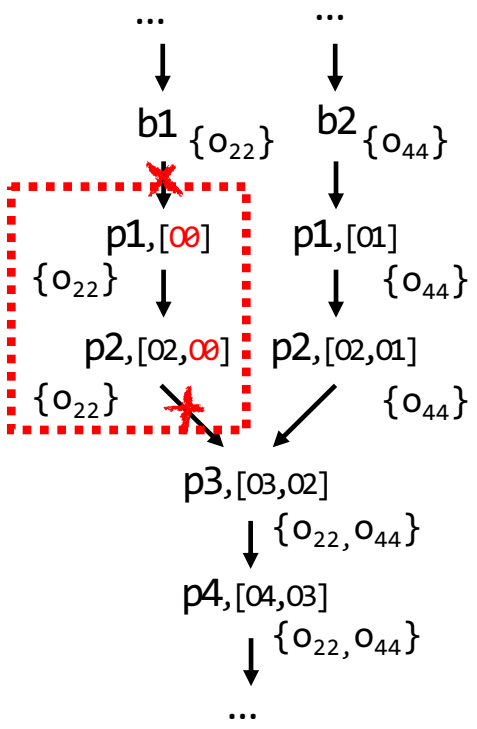
CG



Invalid Elements

{ O0 }

PAG

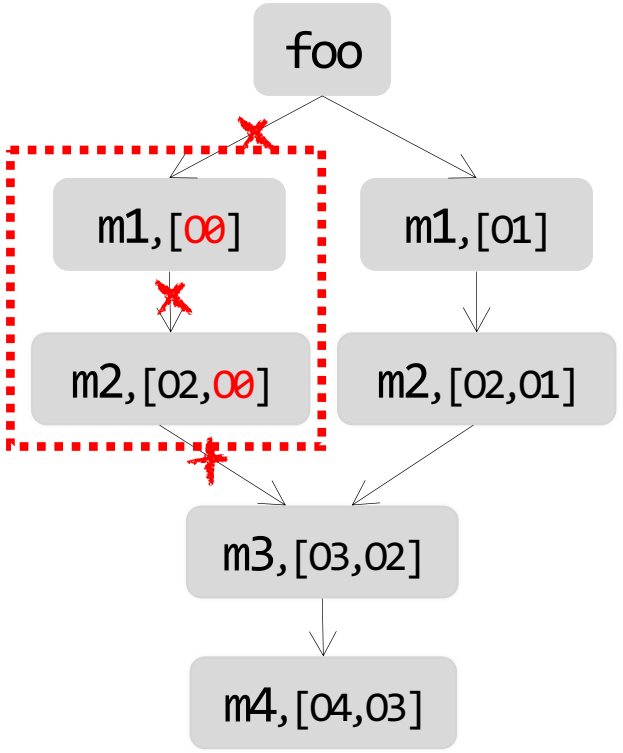


SHARP – Pre-Compute: 2-Obj

Code

```
1  public static void foo() {
2  -  x = new Obj();//O0
3     x.m1(b1);//pts(y)={O0}
4     y.m1(b2);//pts(y)={O1}
5  }
6  public void m1(B p1){
7     a.m2(p1);//pts(a)={O2}
8  }
9  public void m2(B p2) {
10    b.m3(p2);//pts(b)={O3}
11 }
12 public void m3(B p3) {
13    c.m4(p3);//pts(c)={O4}
14 }
15 public void m4(B p4) {
16    ...
17 }
```

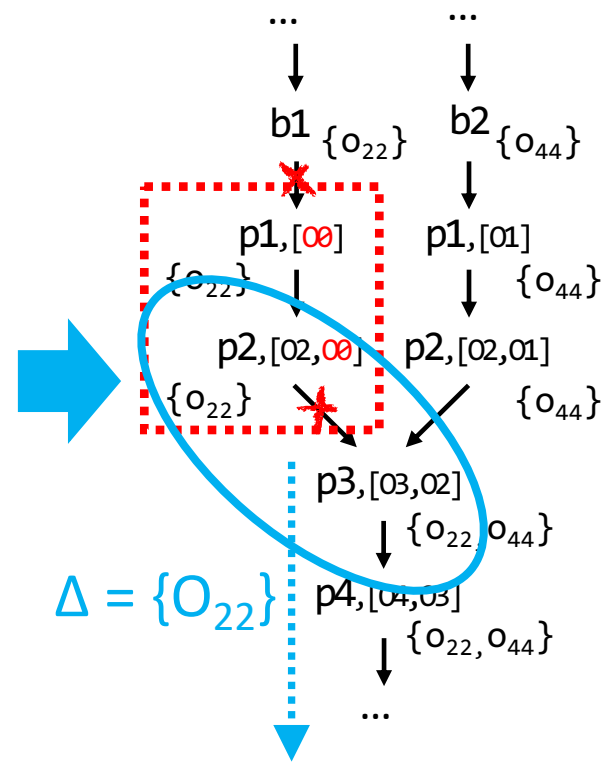
CG



Invalid Elements

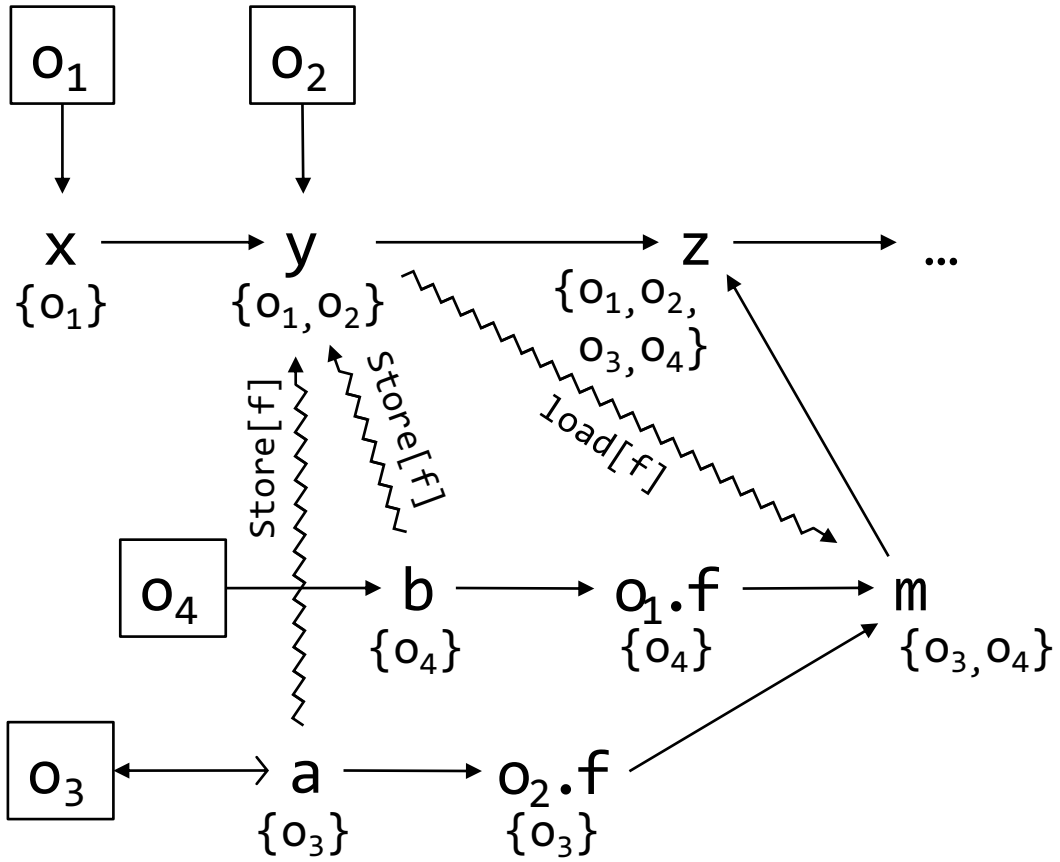
{ O0 }

PAG



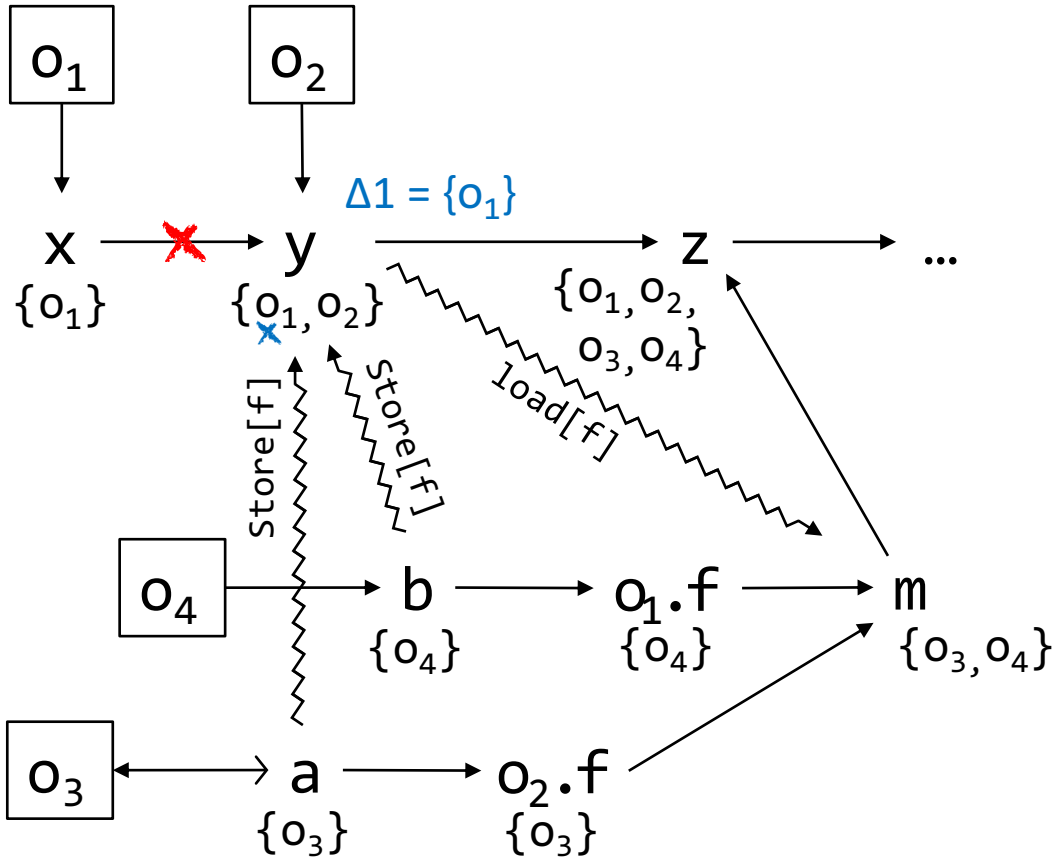
Parallel IPA – One Statement

```
z = y;
if(…) {
   a = new A();//O3
   y = new B();//O2
   y.f = a;
} else {
   b = new A();//O4
   x = new B();//O1
   y = x;
   y.f = b;
}
m = y.f;
z = m;
…
```



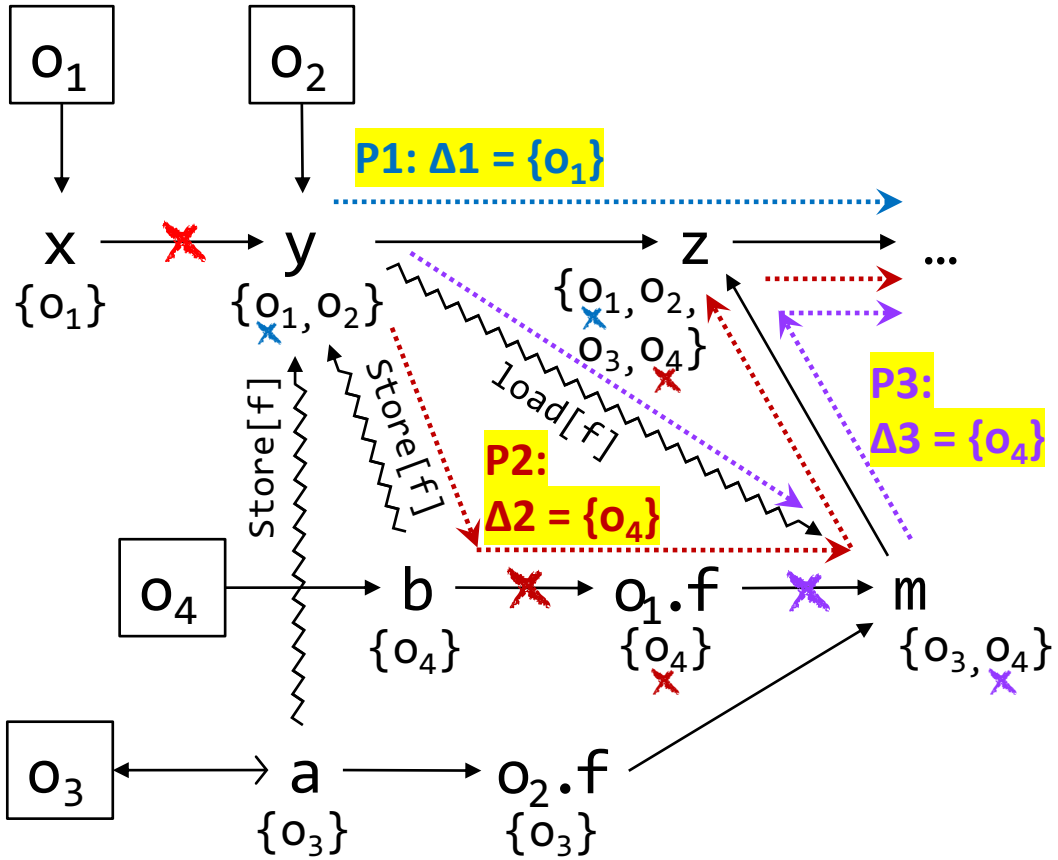
Parallel IPA – One Statement

```
z = y;
if(…) {
   a = new A();//O3
   y = new B();//O2
   y.f = a;
} else {
   b = new A();//O4
   x = new B();//O1
-  y = x;
   y.f = b;
}
m = y.f;
z = m;
…
```



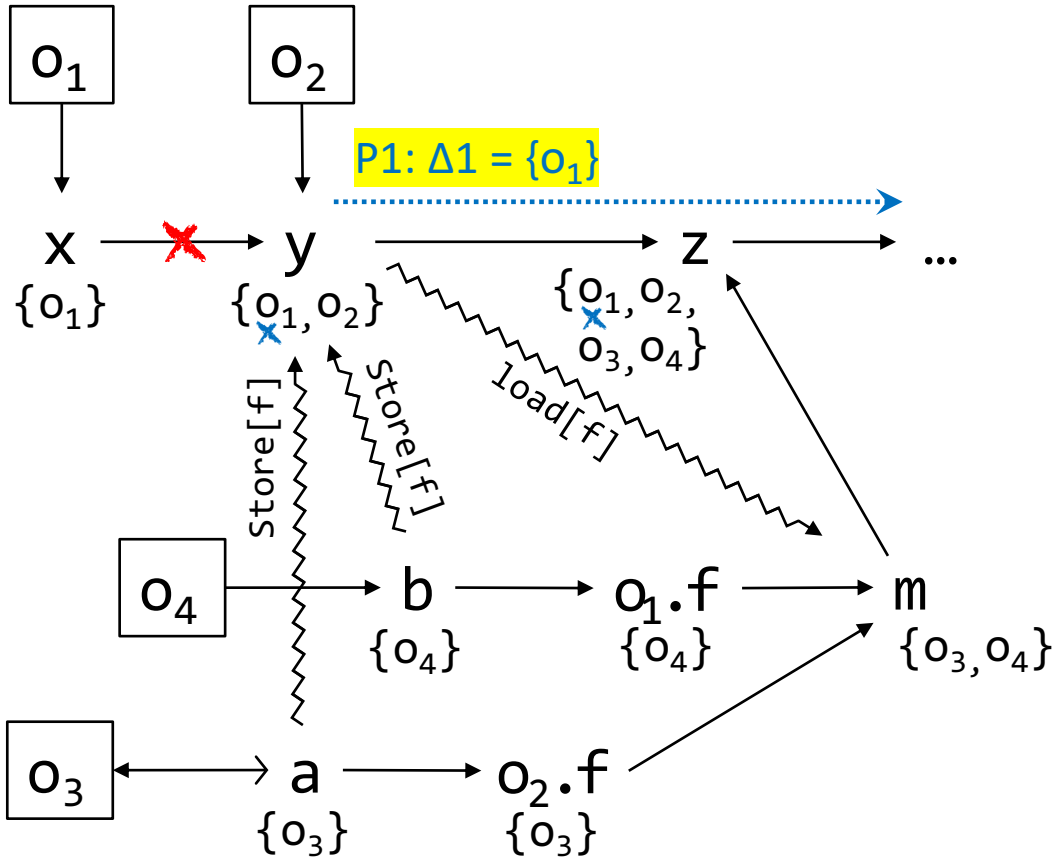
Parallel IPA – One Statement

```
z = y;
if(…) {
   a = new A();//O3
   y = new B();//O2
   y.f = a;
} else {
   b = new A();//O4
   x = new B();//O1
-  y = x;
   y.f = b;
}
m = y.f;
z = m;
…
```



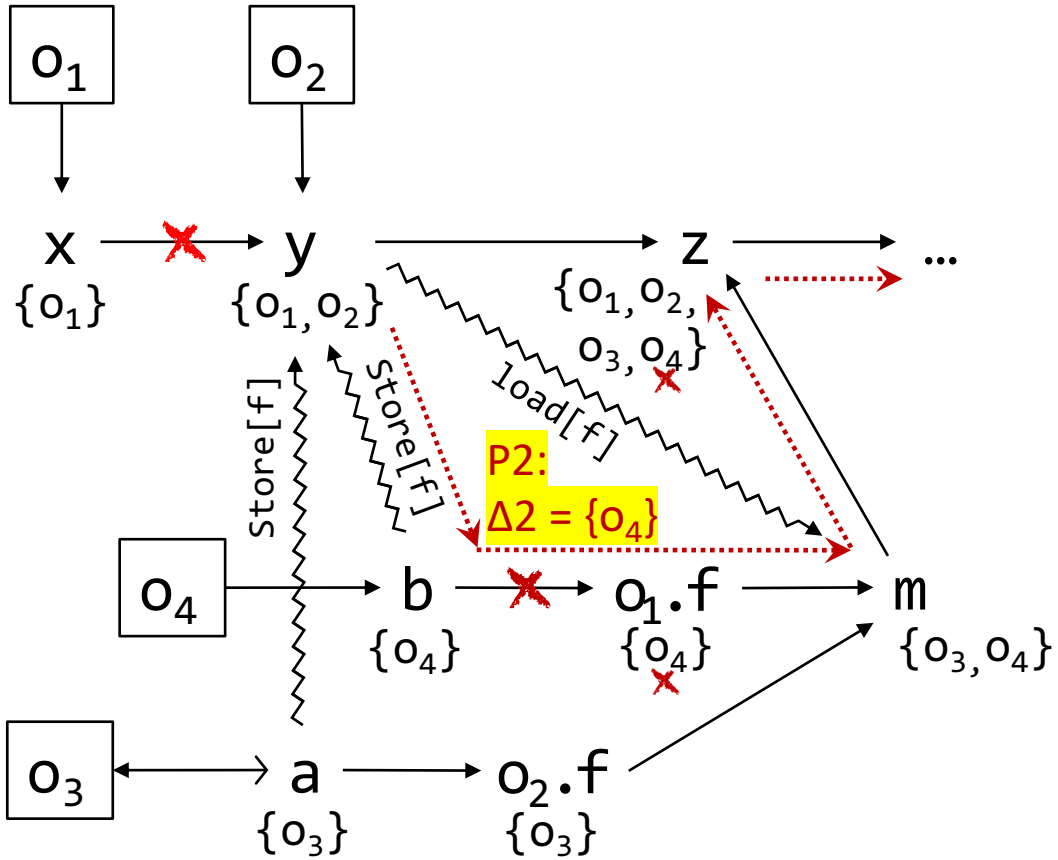
Parallel IPA – One Statement

```
z = y;
if(…) {
   a = new A();//O3
   y = new B();//O2
   y.f = a;
} else {
   b = new A();//O4
   x = new B();//O1
-  y = x;
   y.f = b;
}
m = y.f;
z = m;
…
```



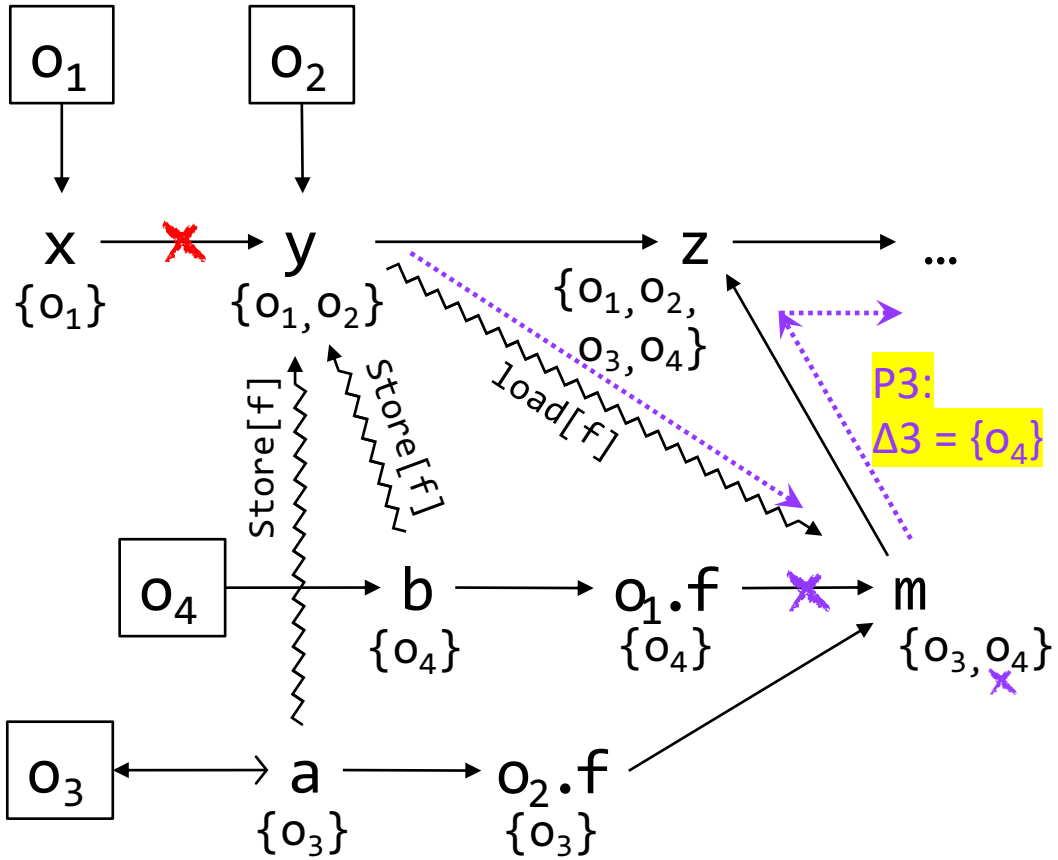
Parallel IPA – One Statement

```
z = y;
if(…) {
   a = new A();//O3
   y = new B();//O2
   y.f = a;
} else {
   b = new A();//O4
   x = new B();//O1
-  y = x;
   y.f = b;
}
m = y.f;
z = m;
…
```



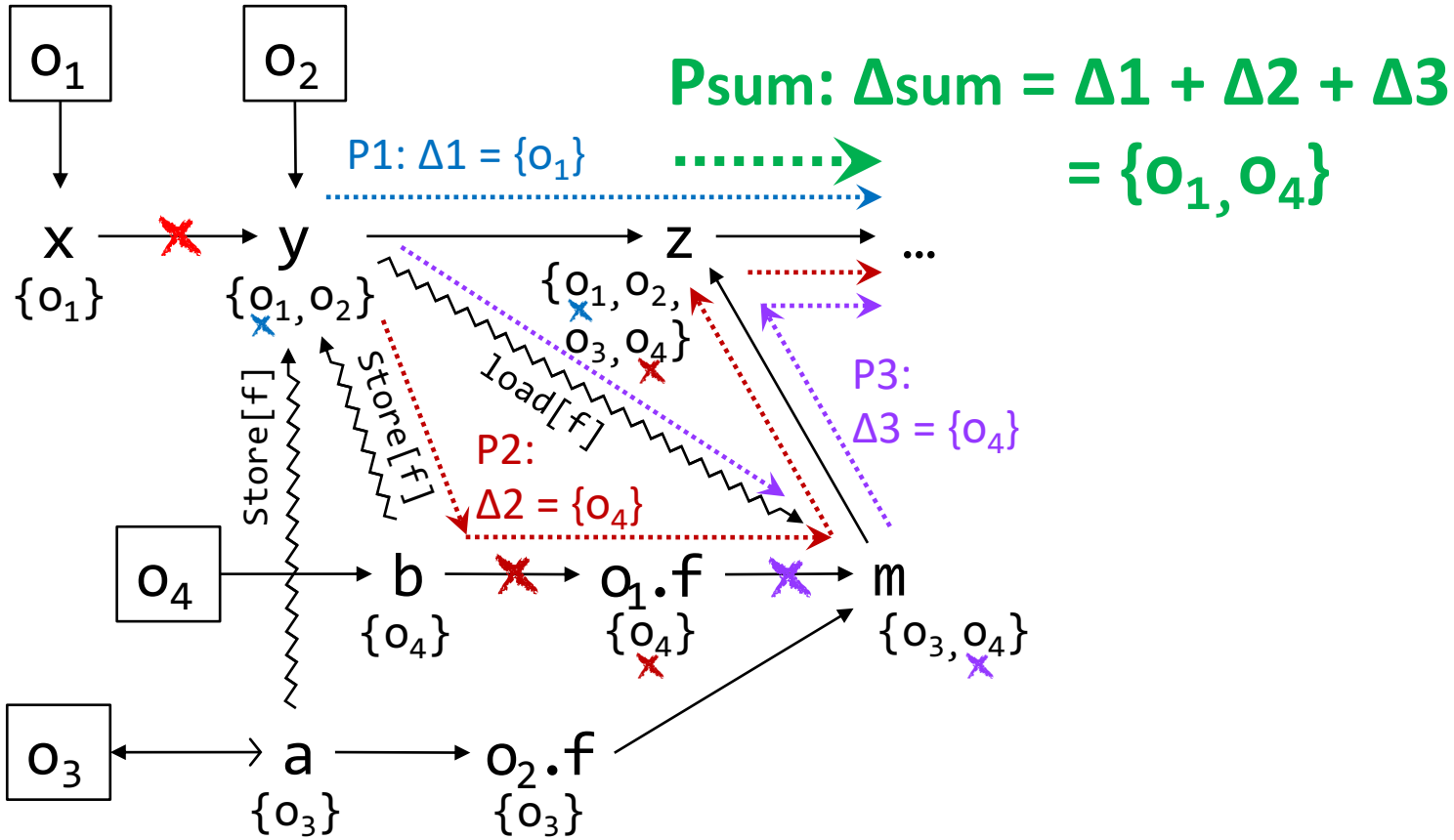
Parallel IPA – One Statement

```
z = y;
if(…) {
   a = new A();//O3
   y = new B();//O2
   y.f = a;
} else {
   b = new A();//O4
   x = new B();//O1
-  y = x;
   y.f = b;
}
m = y.f;
z = m;
…
```



Parallel IPA – One Statement

```
z = y;
if(…) {
   a = new A();//O3
   y = new B();//O2
   y.f = a;
} else {
   b = new A();//O4
   x = new B();//O1
-  y = x;
   y.f = b;
}
m = y.f;
z = m;
…
```

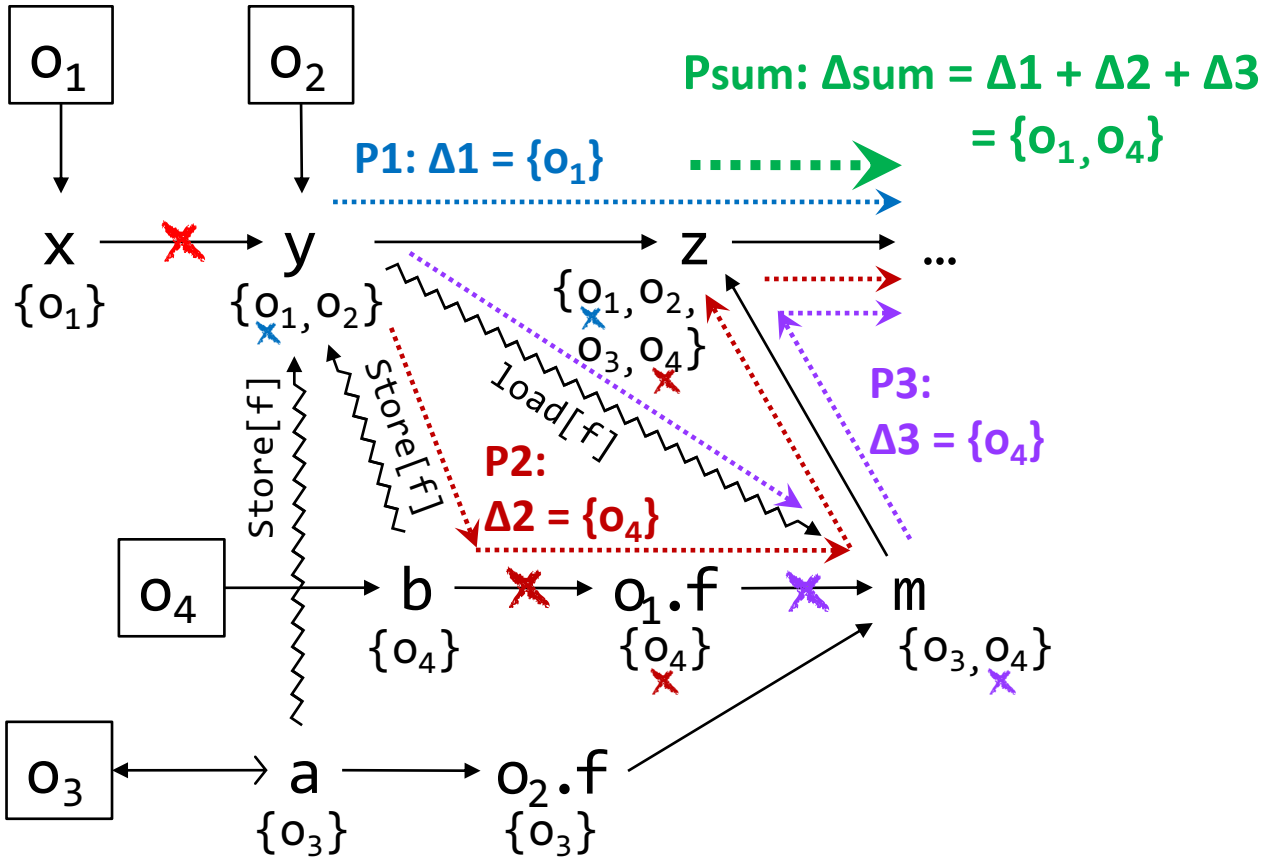


Parallel SHARP – One Statement

Case (i) $P_{sum} = \emptyset$,

Case (ii) $P_{sum} \neq \emptyset \ \wedge \ \Delta_{sum} = \Delta_i \neq \emptyset \ \wedge \ \Delta_j = \emptyset \ (\forall j \in [1, n] \text{ but } j \neq i)$,

Case (iii) $P_{sum} \neq \emptyset \ \wedge \ \Delta_{sum} = \bigcup_{i=1}^{k} \Delta_i \ \wedge \ \Delta_i \neq \emptyset \ (1 < k \leq n)$.

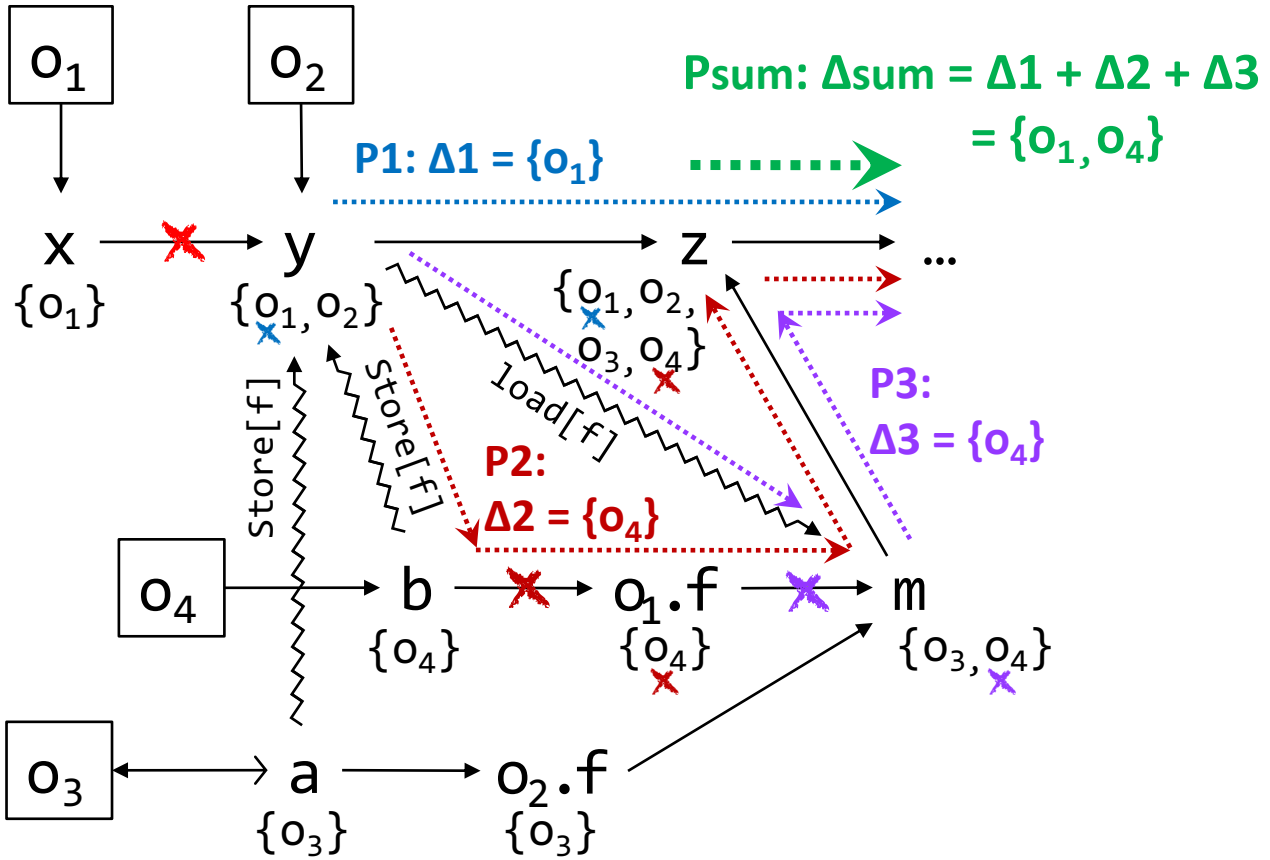


Parallel SHARP – One Statement

Case (i) $P_{sum} = \emptyset$,

Case (ii) $P_{sum} \neq \emptyset \ \wedge \ \Delta_{sum} = \Delta_i \neq \emptyset \ \wedge \ \Delta_j = \emptyset \ (\forall j \in [1, n] \text{ but } j \neq i)$,

Case (iii) $P_{sum} \neq \emptyset \ \wedge \ \Delta_{sum} = \bigcup_{i=1}^{k} \Delta_i \ \wedge \ \Delta_i \neq \emptyset \ (1 < k \leq n)$. $\begin{cases} \Delta_i \bigcap \Delta_j = \emptyset \\ \Delta_i \bigcap \Delta_j = \delta \neq \emptyset \end{cases}$

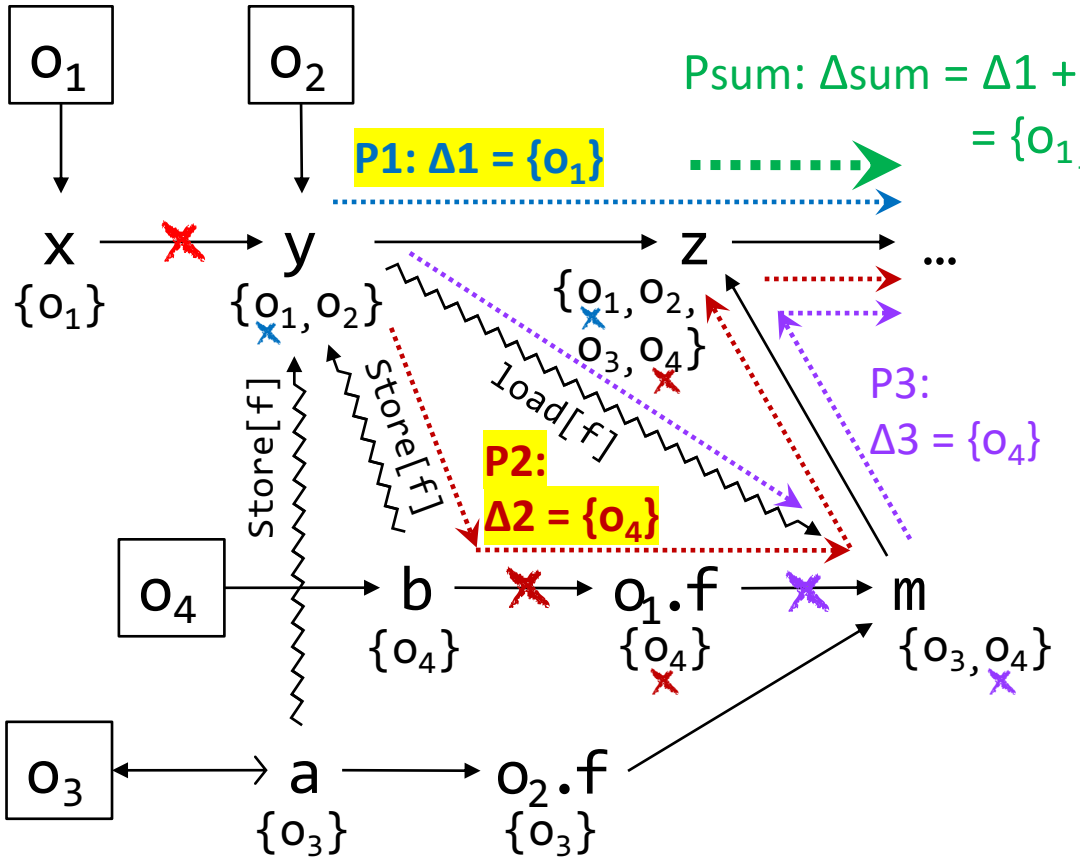


Parallel SHARP – One Statement

Case (i) $P_{sum} = \emptyset$,

Case (ii) $P_{sum} \neq \emptyset \ \wedge \ \Delta_{sum} = \Delta_i \neq \emptyset \ \wedge \ \Delta_j = \emptyset \ (\forall j \in [1, n] \text{ but } j \neq i)$,

Case (iii) $P_{sum} \neq \emptyset \ \wedge \ \Delta_{sum} = \bigcup_{i=1}^{k} \Delta_i \ \wedge \ \Delta_i \neq \emptyset \ (1 < k \leq n)$. $\begin{cases} \Delta_i \bigcap \Delta_j = \emptyset \\ \Delta_i \bigcap \Delta_j = \delta \neq \emptyset \end{cases}$



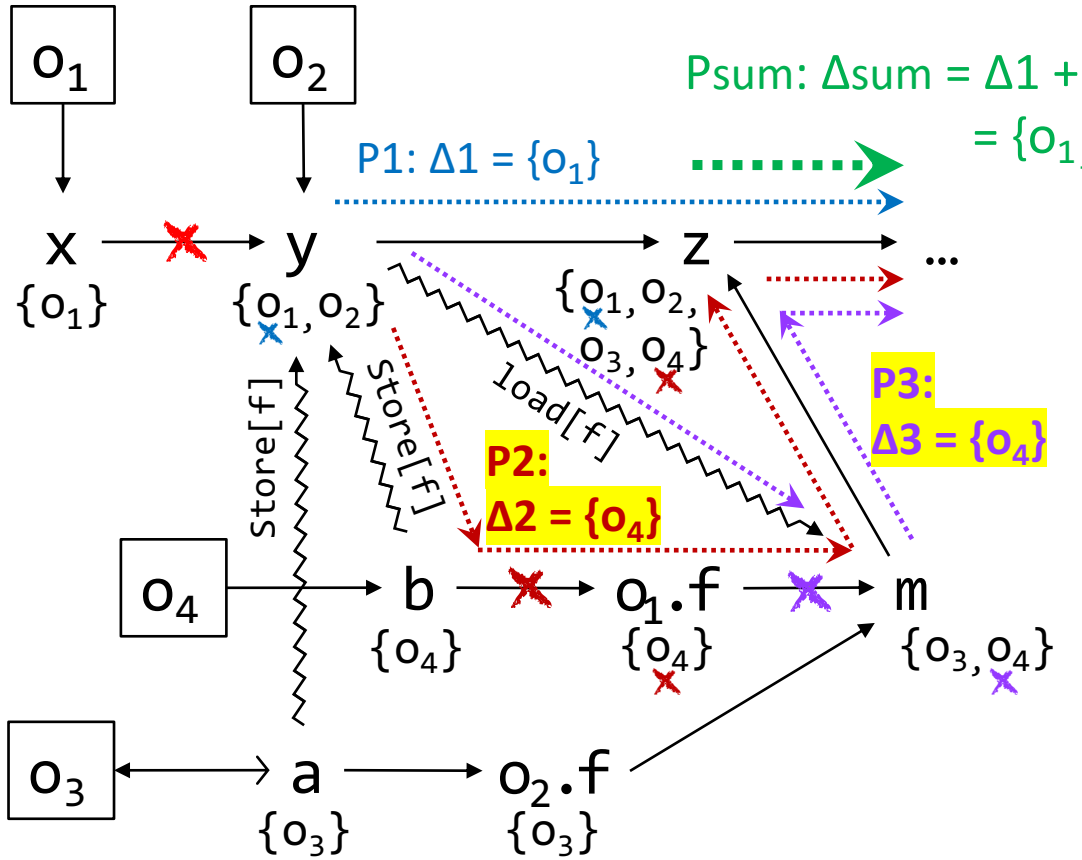
Δ1 ∩ Δ2 = {o1} ∩ {o4} = ∅

Parallel SHARP – One Statement

Case (i) $P_{sum} = \emptyset$,

Case (ii) $P_{sum} \neq \emptyset \ \wedge \ \Delta_{sum} = \Delta_i \neq \emptyset \ \wedge \ \Delta_j = \emptyset \ (\forall j \in [1, n] \text{ but } j \neq i)$,

Case (iii) $P_{sum} \neq \emptyset \ \wedge \ \Delta_{sum} = \bigcup_{i=1}^{k} \Delta_i \ \wedge \ \Delta_i \neq \emptyset \ (1 < k \leq n)$. $\begin{cases} \Delta_i \bigcap \Delta_j = \emptyset \\ \Delta_i \bigcap \Delta_j = \delta \neq \emptyset \end{cases}$



Δ2 ∩ Δ3 = δ
= {o4} ∩ {o4} = {o4} ≠ ∅

Parallel SHARP

- Intraprocedural Code Changes.

✓ • One statement change

✗ • Multiple statement changes

- Interprocedural Code Changes

✓ • One statement change in each method (either deletion or addition)

✗ • Multiple statement changes in each method

✗ • Statement Changes with Both Deletion and Addition

SHARP - Performance

- Empirical Evaluation
  - Real-world Git repos
    - 1M LOC
    - > 10 years
  - 10 Git commits
    - #Add: 1386
    - #Delete: 900
  - SHARP
    - k-CFA
    - k-Obj

Project	Pointer Analysis	Full Time	Incremental Avg. Time		Statistics of PAG		
			Per Commit		#Pointer	#Object	#Edge
Hbase	CI	2.06min	IPA	6494.1ms/18.1x	310,155	22,327	228,889,050
	1-CFA	2.65min	SHARP(Max)	11452.94ms/12.9x	642,685	22,987	110,152,245
	2-CFA	3.56h		182858.63ms/69.2x	4,594,120	38,607	703,547,087
	1-obj	13.26s		716.15ms/17.5x	70,060	4,861	2,098,257
	2-obj	17.07s		958.07ms/16.8x	102,526	5,712	3,803,780
Lucene	CI	9.68s	IPA	21.5ms/449.1x	127,596	7,418	5,265,380
	1-CFA	62.67s	SHARP(Max)	2631.47ms/22.8x	405,498	14,691	79,307,176
	2-CFA	7.63h		37555.68ms/731.0x	3,990,653	33,271	1,094,819,728
	1-obj	25.84s		901.41ms/27.6x	116,581	6,420	8,515,319
	2-obj	30.88s		311.62ms/98.1x	143,353	8,028	12,395,079
Yarn	CI	22.96s	IPA	406.2ms/55.5x	145,734	11,278	18,660,434
	1-CFA	44.97s	SHARP(Max)	2062.05ms/20.8x	310,889	10,196	16,734,763
	2-CFA	2.47h		11948.81ms/742.8x	3,714,158	20,493	370,207,476
	1-obj	50.62s		320.36ms/157.0x	152,089	7,580	11,071,426
	2-obj	52min		18637.75ms/165.8x	616,034	34,472	1,091,507,725
Zookeeper	CI	10.36s	IPA	601.3ms/16.2x	96,238	8,705	9,308,586
	1-CFA	26.11s	SHARP(Max)	4495.83ms/4.81x	301,099	9,491	33,759,731
	2-CFA	6.35h		15517.79ms/880.6x	3,925,385	26,220	1,092,490,217
	1-obj	15.17s		1243.04ms/11.2x	79,795	4,297	2,817,254
	2-obj	18.58s		1710.73ms/9.8x	90,276	5,331	5,295,305
Average		1.15h	IPA	1783.95ms/193.5x			
			SHARP(Max)	18332.64ms/186.8x			

SHARP - Performance

- Empirical Evaluation
  - Real-world Git repos
    - 1M LOC
    - > 10 years
  - 10 Git commits
    - #Add: 1386
    - #Delete: 900
  - SHARP
    - k-CFA
    - k-Obj

Project	Pointer Analysis	Full Time	Incremental Avg. Time Per Commit		Statistics of PAG #Pointer	#Object	#Edge
Hbase	CI	**2.06min**	IPA	6494.1ms/18.1x	**310,155**	**22,327**	**228,889,050**
	1-CFA	**2.65min**	SHARP(Max)	11452.94ms/12.9x	**642,685**	**22,987**	**110,152,245**
	2-CFA	**3.56h**		182858.63ms/69.2x	**4,594,120**	**38,607**	**703,547,087**
	1-obj	**13.26s**		716.15ms/17.5x	**70,060**	**4,861**	**2,098,257**
	2-obj	**17.07s**		958.07ms/16.8x	**102,526**	**5,712**	**3,803,780**
Lucene	CI	**9.68s**	IPA	21.5ms/449.1x	**127,596**	**7,418**	**5,265,380**
	1-CFA	**62.67s**	SHARP(Max)	2631.47ms/22.8x	**405,498**	**14,691**	**79,307,176**
	2-CFA	**7.63h**		37555.68ms/731.0x	**3,990,653**	**33,271**	**1,094,819,728**
	1-obj	**25.84s**		901.41ms/27.6x	**116,581**	**6,420**	**8,515,319**
	2-obj	**30.88s**		311.62ms/98.1x	**143,353**	**8,028**	**12,395,079**
Yarn	CI	**22.96s**	IPA	406.2ms/55.5x	**145,734**	**11,278**	**18,660,434**
	1-CFA	**44.97s**	SHARP(Max)	2062.05ms/20.8x	**310,889**	**10,196**	**16,734,763**
	2-CFA	**2.47h**		11948.81ms/742.8x	**3,714,158**	**20,493**	**370,207,476**
	1-obj	**50.62s**		320.36ms/157.0x	**152,089**	**7,580**	**11,071,426**
	2-obj	**52min**		18637.75ms/165.8x	**616,034**	**34,472**	**1,091,507,725**
Zookeeper	CI	**10.36s**	IPA	601.3ms/16.2x	**96,238**	**8,705**	**9,308,586**
	1-CFA	**26.11s**	SHARP(Max)	4495.83ms/4.81x	**301,099**	**9,491**	**33,759,731**
	2-CFA	**6.35h**		15517.79ms/880.6x	**3,925,385**	**26,220**	**1,092,490,217**
	1-obj	**15.17s**		1243.04ms/11.2x	**79,795**	**4,297**	**2,817,254**
	2-obj	**18.58s**		1710.73ms/9.8x	**90,276**	**5,331**	**5,285,365**
Average		1.15h	IPA	1783.95ms/193.5x			
			SHARP(Max)	18332.64ms/186.8x			

SHARP - Performance

- Empirical Evaluation
  - Real-world Git repos
    - 1M LOC
    - > 10 years
  - 10 Git commits
    - #Add: 1386
    - #Delete: 900
  - SHARP
    - k-CFA
    - k-Obj

Project	Pointer Analysis	Full Time	Incremental Avg. Time		Statistics of PAG		
			Per Commit		#Pointer	#Object	#Edge
Hbase	CI	2.06min	IPA	6494.1ms/18.1x	310,155	22,327	228,889,050
	1-CFA	2.65min	SHARP(Max)	11452.94ms/12.9x	642,685	22,987	110,152,245
	2-CFA	3.56h		182858.63ms/69.2x	4,594,120	38,607	703,547,087
	1-obj	13.26s		716.15ms/17.5x	70,060	4,861	2,098,257
	2-obj	17.07s		958.07ms/16.8x	102,526	5,712	3,803,780
Lucene	CI	9.68s	IPA	21.5ms/449.1x	127,596	7,418	5,265,380
	1-CFA	62.67s	SHARP(Max)	2631.47ms/22.8x	405,498	14,691	79,307,176
	2-CFA	7.63h		37555.68ms/731.0x	3,990,653	33,271	1,094,819,728
	1-obj	25.84s		901.41ms/27.6x	116,581	6,420	8,515,319
	2-obj	30.88s		311.62ms/98.1x	143,353	8,028	12,395,079
Yarn	CI	22.96s	IPA	406.2ms/55.5x	145,734	11,278	18,660,434
	1-CFA	44.97s	SHARP(Max)	2062.05ms/20.8x	310,889	10,196	16,734,763
	2-CFA	2.47h		11948.81ms/742.8x	3,714,158	20,493	370,207,476
	1-obj	50.62s		320.36ms/157.0x	152,089	7,580	11,071,426
	2-obj	52min		18637.75ms/165.8x	616,034	34,472	1,091,507,725
Zookeeper	CI	10.36s	IPA	601.3ms/16.2x	96,238	8,705	9,308,586
	1-CFA	26.11s	SHARP(Max)	4495.83ms/4.81x	301,099	9,491	33,759,731
	2-CFA	6.35h		15517.79ms/880.6x	3,925,385	26,220	1,092,490,217
	1-obj	15.17s		1243.04ms/11.2x	79,795	4,297	2,817,254
	2-obj	18.58s		1710.73ms/9.8x	90,276	5,331	5,295,305
Average		1.15h	**IPA**	**1783.95ms/193.5x**			
			SHARP(Max)	**18332.64ms/186.8x**			

SHARP - Performance

- Empirical Evaluation
  - Real-world Git repos
    - 1M LOC
    - > 10 years
  - 10 Git commits
    - #Add: 1386
    - #Delete: 900
  - SHARP
    - k-CFA
    - k-Obj

Project	Pointer Analysis	Full Time	Incremental Avg. Time		Statistics of PAG		
			Per Commit		#Pointer	#Object	#Edge
Hbase	CI	2.06min	IPA	6494.1ms/**18.1x**	310,155	22,327	228,889,050
	1-CFA	2.65min	SHARP(Max)	11452.94ms/**12.9x**	642,685	22,987	110,152,245
	2-CFA	3.56h		182858.63ms/**69.2x**	4,594,120	38,607	703,547,087
	1-obj	13.26s		716.15ms/**17.5x**	70,060	4,861	2,098,257
	2-obj	17.07s		958.07ms/**16.8x**	102,526	5,712	3,803,780
Lucene	CI	9.68s	IPA	21.5ms/**449.1x**	127,596	7,418	5,265,380
	1-CFA	62.67s	SHARP(Max)	2631.47ms/**22.8x**	405,498	14,691	79,307,176
	2-CFA	7.63h		37555.68ms/**731.0x**	3,990,653	33,271	1,094,819,728
	1-obj	25.84s		901.41ms/**27.6x**	116,581	6,420	8,515,319
	2-obj	30.88s		311.62ms/**98.1x**	143,353	8,028	12,395,079
Yarn	CI	22.96s	IPA	406.2ms/**55.5x**	145,734	11,278	18,660,434
	1-CFA	44.97s	SHARP(Max)	2062.05ms/**20.8x**	310,889	10,196	16,734,763
	2-CFA	2.47h		11948.81ms/**742.8x**	3,714,158	20,493	370,207,476
	1-obj	50.62s		320.36ms/**157.0x**	152,089	7,580	11,071,426
	2-obj	52min		18637.75ms/**165.8x**	616,034	34,472	1,091,507,725
Zookeeper	CI	10.36s	IPA	601.3ms/**16.2x**	96,238	8,705	9,308,586
	1-CFA	26.11s	SHARP(Max)	4495.83ms/**4.81x**	301,099	9,491	33,759,731
	2-CFA	6.35h		15517.79ms/**880.6x**	3,925,385	26,220	1,092,490,217
	1-obj	15.17s		1243.04ms/**11.2x**	79,795	4,297	2,817,254
	2-obj	18.58s		1710.73ms/**9.8x**	90,276	5,331	5,295,305
Average		1.15h	IPA	1783.95ms/193.5x			
			SHARP(Max)	18332.64ms/186.8x			

Thank You!

https://github.com/april1989/Incremental_Points_to_Analysis/tree/master/Sharp