Data-Driven Context-Sensitivity for Points-to Analysis

Sehun Jeong*, Minseok Jeon*,
Sungdeok Cha, Hakjoo Oh

KOREA UNIVERSITY

OOPSLA 2017

Static analyzer

Heuristics in Static Analysis

- Modern static analyzers use many heuristics
  - Which procedures should be analyzed context-sensitively?
  - Which variable should be analyzed flow-sensitively?
  - Which program parts to analyze soundly or unsoundly?,etc
- Developing a good heuristic is an art
  - Empirically done by analysis designers: nontrivial & suboptimal

Automatically Generating Heuristics from Data

- Automate the process: use data to make heuristic decisions in static analysis

GitHub

Context-sensitive heuristics
Flow-sensitive heuristics
Unsoundness heuristics
...

- **Automatic:** little reliance on analysis designer
- **Powerful:** machine-tuned outperforms hand-tunning
- **Stable:** can be generalized for target programs

Automatically Generating Heuristics from Data

- Automate the process: use data to make heuristic decisions in static analysis

GitHub →

Context-sensitive heuristics
Flow-sensitive heuristics
Unsoundness heuristics
...

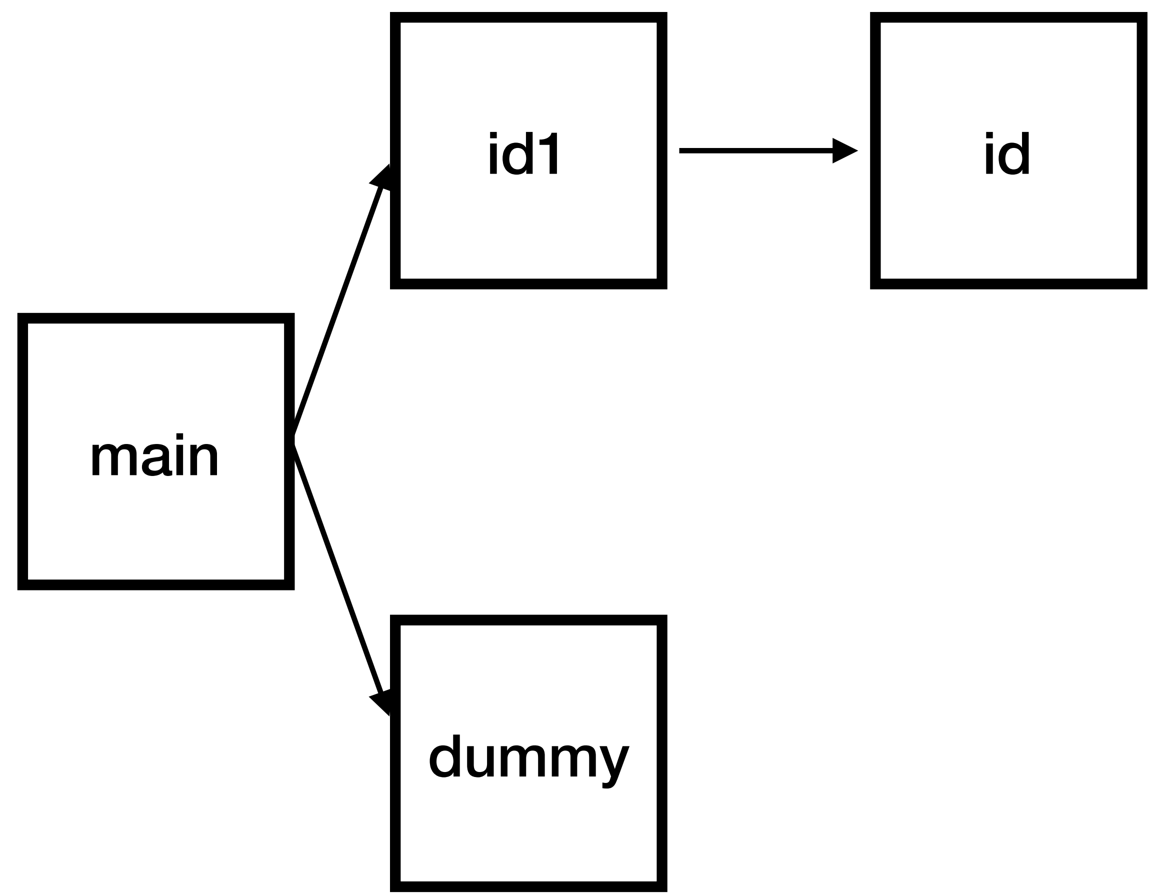
- **Automatic:** little reliance on analysis designer
- **Powerful:** machine-tuned outperforms hand-tunning
- **Stable:** can be generalized for target programs

Context insensitive

```java
class A{} class B{}
class C{
  public static Object id(Object v){
    return v;}
  public static Object id1(Object v1){
    return id(v1);}
  public static void dummy(){}

  public static void main(){
    A a = (A)id1(new A()); //query
    B b = (B)id1(new B()); //query
    dummy();
    dummy();
}}
```

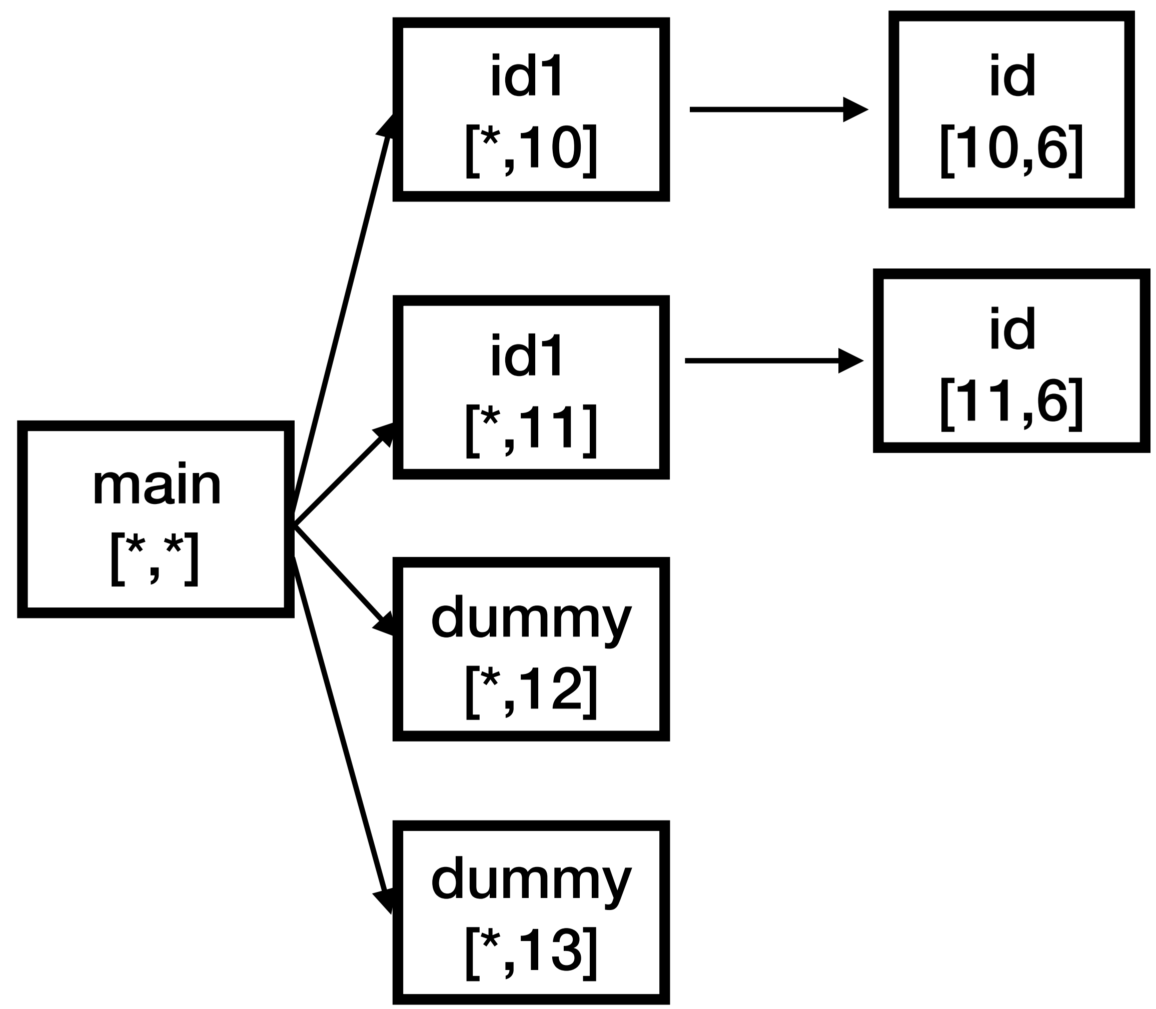
Without context-sensitivity, analysis fails to prove queries



2-call-site sensitive

```
1:  class A{} class B{}
2:  class C{
3:    public static Object id(Object v){
4:      return v;}
5:    public static Object id1(Object v1){
6:      return id(v1);}
7:    public static void dummy(){}
8:
9:    public static void main(){
10:     A a = (A)id1(new A()); //query
11:     B b = (B)id1(new B()); //query
12:     dummy();
13:     dummy();
14: }}
```

2-call-site sensitivity succeed but not scale

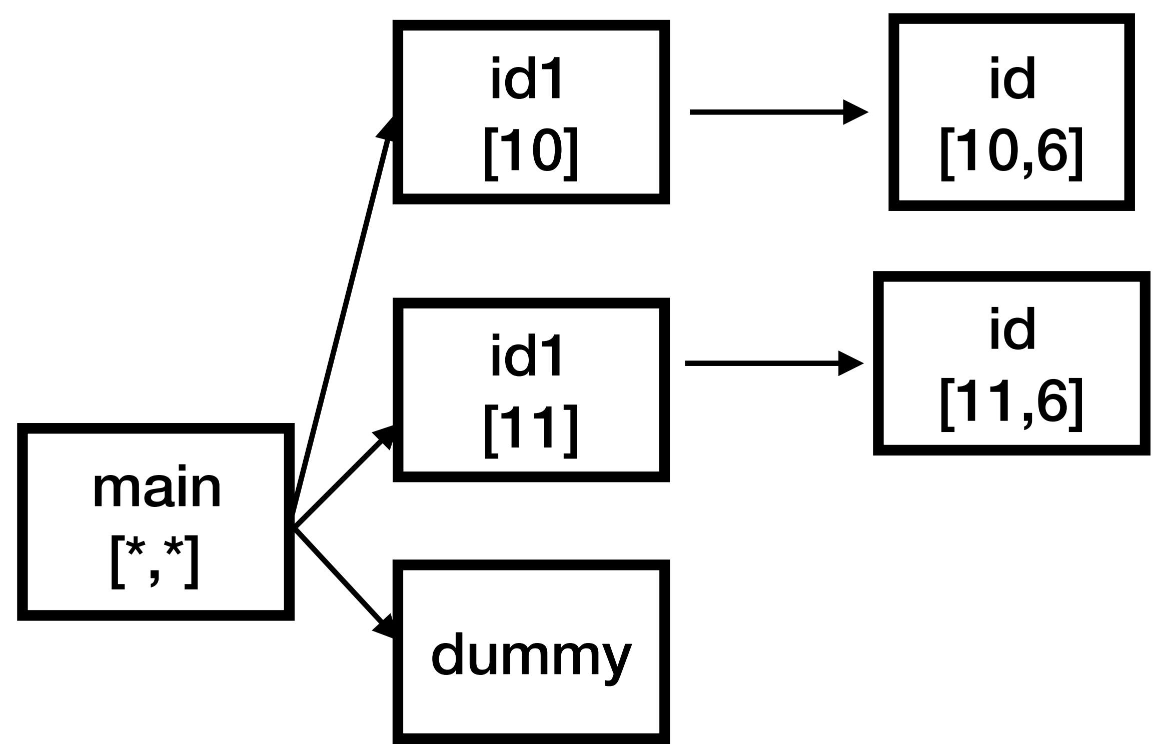


Selective Context Sensitivity

```
class A{} class B{}
class C{
  public static Object id(Object v){
    return v;}
  public static Object id1(Object v1){
    return id(v1);}
  public static void dummy(){}

  public static void main(){
    A a = (A)id1(new A()); //query
    B b = (B)id1(new B()); //query
    dummy();
    dummy();
}}
```

Apply 2-call-sens: {id}
Apply 1-call-sens: {id1}
Apply insens: {dummy}



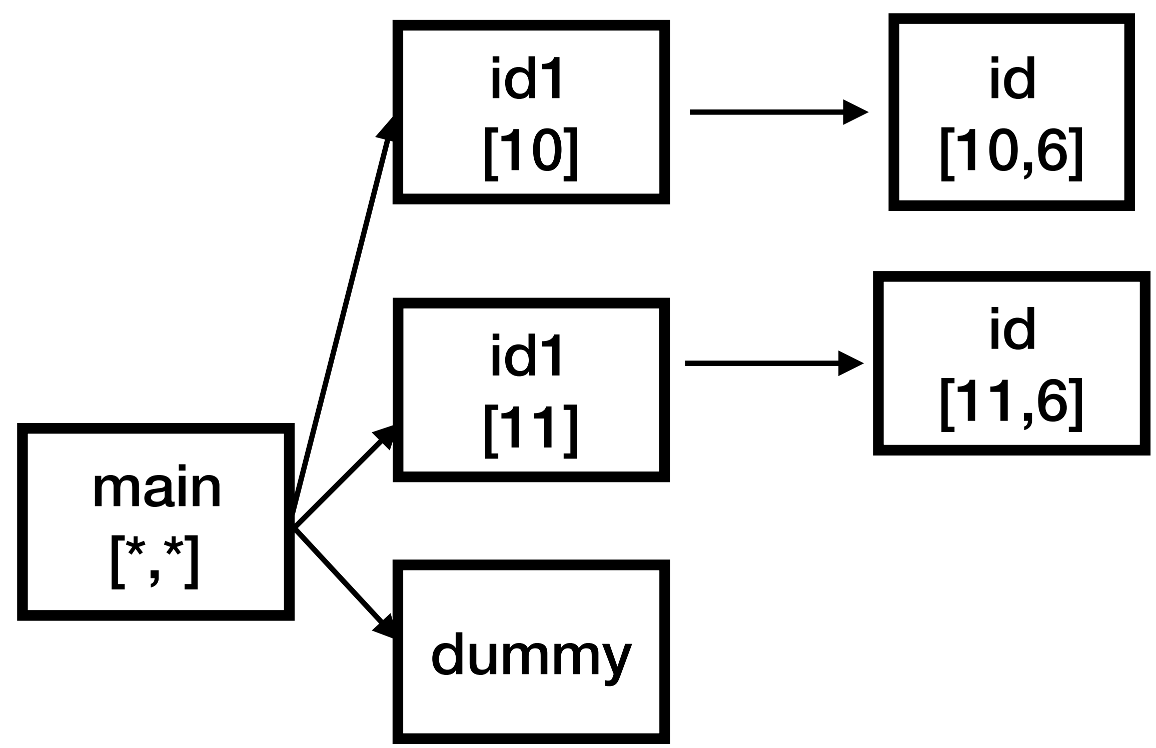
Selective Context Sensitivity

Challenge: How to decide? Data-driven approach

```
class A{} class B{}
class C{
  public static Object id(Object v){
    return v;}
  public static Object id1(Object v1){
    return id(v1);}
  public static void dummy(){}

  public static void main(){
    A a = (A)id1(new A()); //query
    B b = (B)id1(new B()); //query
    dummy();
    dummy();
}}
```

Apply 2-call-sens: {id}
Apply 1-call-sens: {id1}
Apply insens: {dummy}



Data-Driven Ctx-Sensitivity

Parametric static analyzer

Train data (Programs)

atomic features (a1, a2, …, a25)

e.g., methods have if stmt, methods return void, method have invocation stmt, etc

Our DD Framework

Data-Driven Ctx-Sensitivity

Parametric static analyzer

Train data (Programs)

atomic features (a1, a2, ..., a25)

e.g., methods have if stmt, methods return void, method have invocation stmt, etc

Our DD Framework

- Methods that require 2-hybrid object-sensitivity

$$1 \wedge \neg 3 \wedge \neg 6 \wedge 8 \wedge \neg 9 \wedge \neg 16 \wedge \neg 17 \wedge \neg 18 \wedge \neg 19 \wedge \neg 20 \wedge \ldots \wedge \neg 24 \wedge \neg 25$$

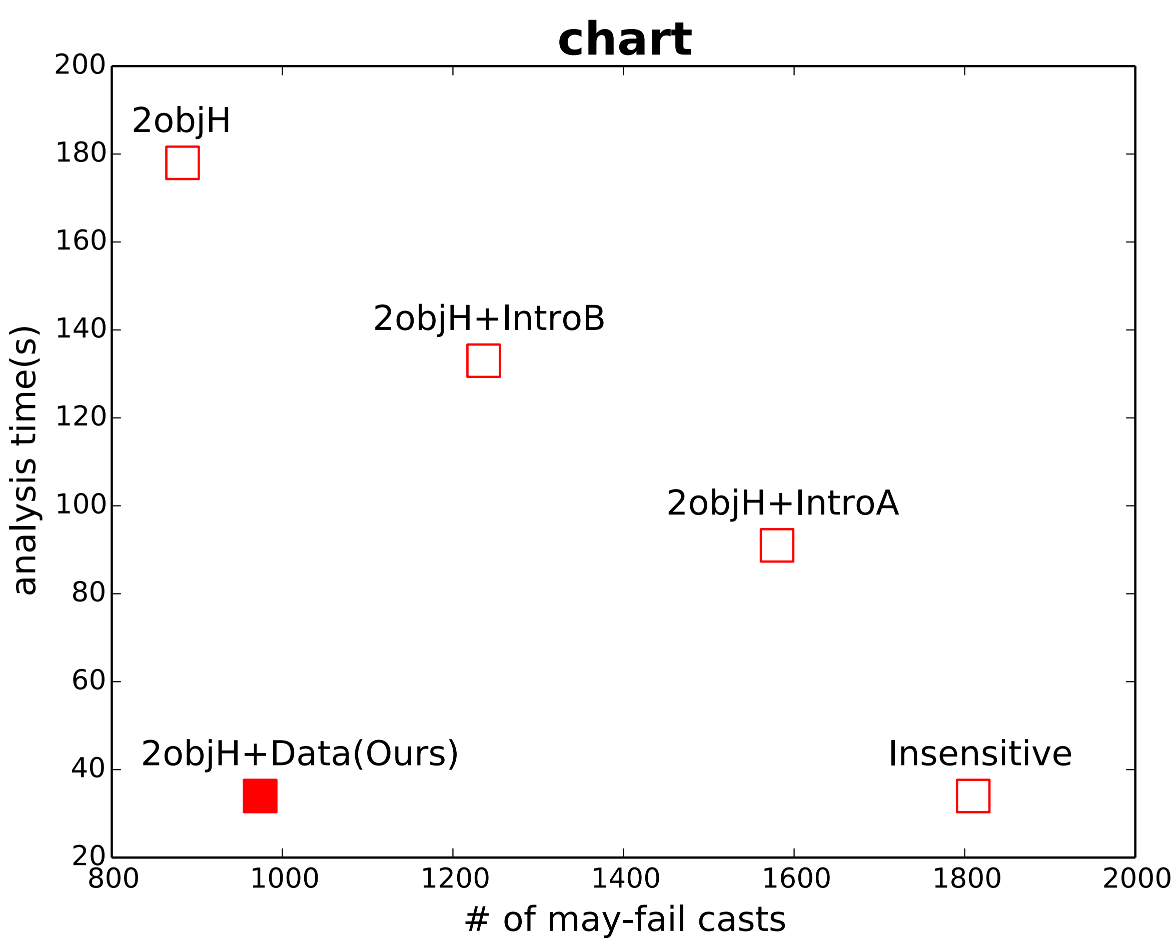
- Methods that require 1-hybrid object-sensitivity

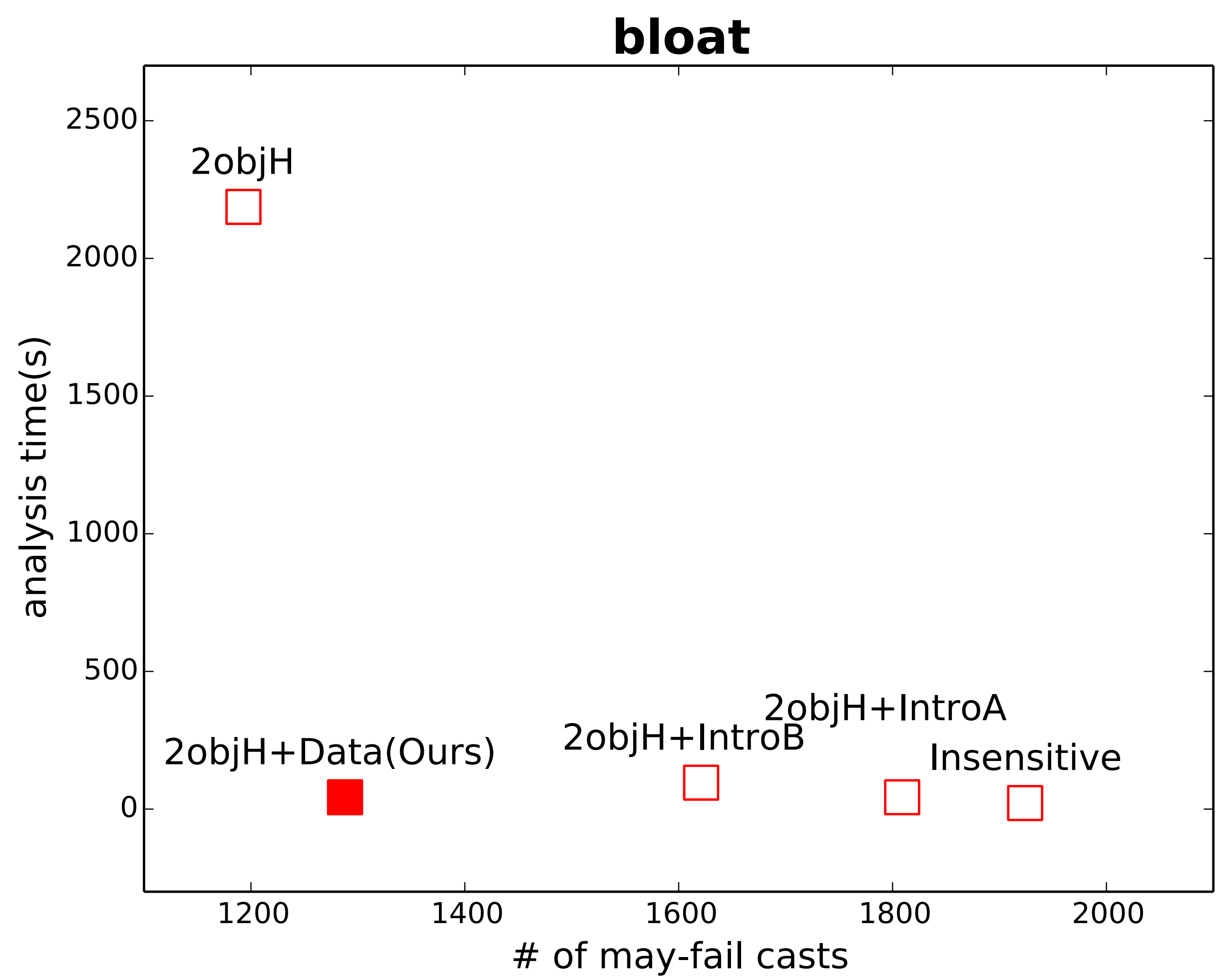
$$(1 \wedge \neg 3 \wedge \neg 4 \wedge \neg 7 \wedge \neg 8 \wedge 6 \wedge \neg 9 \wedge \neg 15 \wedge \neg 16 \wedge \neg 17 \wedge \ldots \wedge \neg 24 \wedge \neg 25) \vee$$
$$(\neg 3 \wedge \neg 4 \wedge \neg 7 \wedge \neg 8 \wedge \neg 9 \wedge 10 \wedge 11 \wedge 12 \wedge 13 \wedge \neg 16 \wedge \ldots \wedge \neg 24 \wedge \neg 25) \vee$$
$$(\neg 3 \wedge \neg 9 \wedge 13 \wedge 14 \wedge 15 \wedge \neg 16 \wedge \neg 17 \wedge \neg 18 \wedge \neg 19 \wedge \ldots \wedge \neg 24 \wedge \neg 25) \vee$$
$$(1 \wedge 2 \wedge \neg 3 \wedge 4 \wedge \neg 5 \wedge \neg 6 \wedge \neg 7 \wedge \neg 8 \wedge \neg 9 \wedge \neg 10 \wedge \neg 13 \wedge \ldots \wedge \neg 24 \wedge \neg 25)$$

Performance

- Training with 4 small programs from DaCapo, and applied to 6 large programs (1 for validation)

- Machine-tuning outperforms hand-tuning

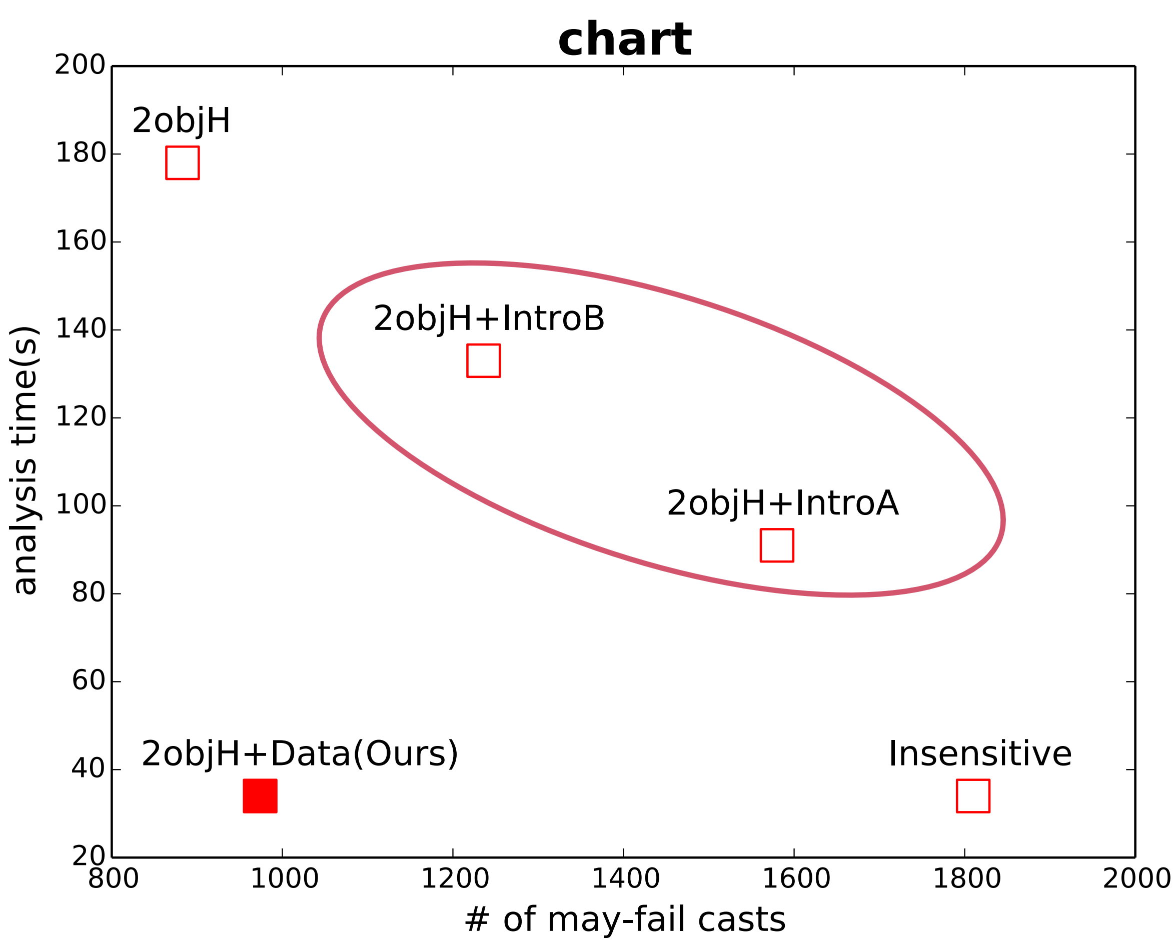


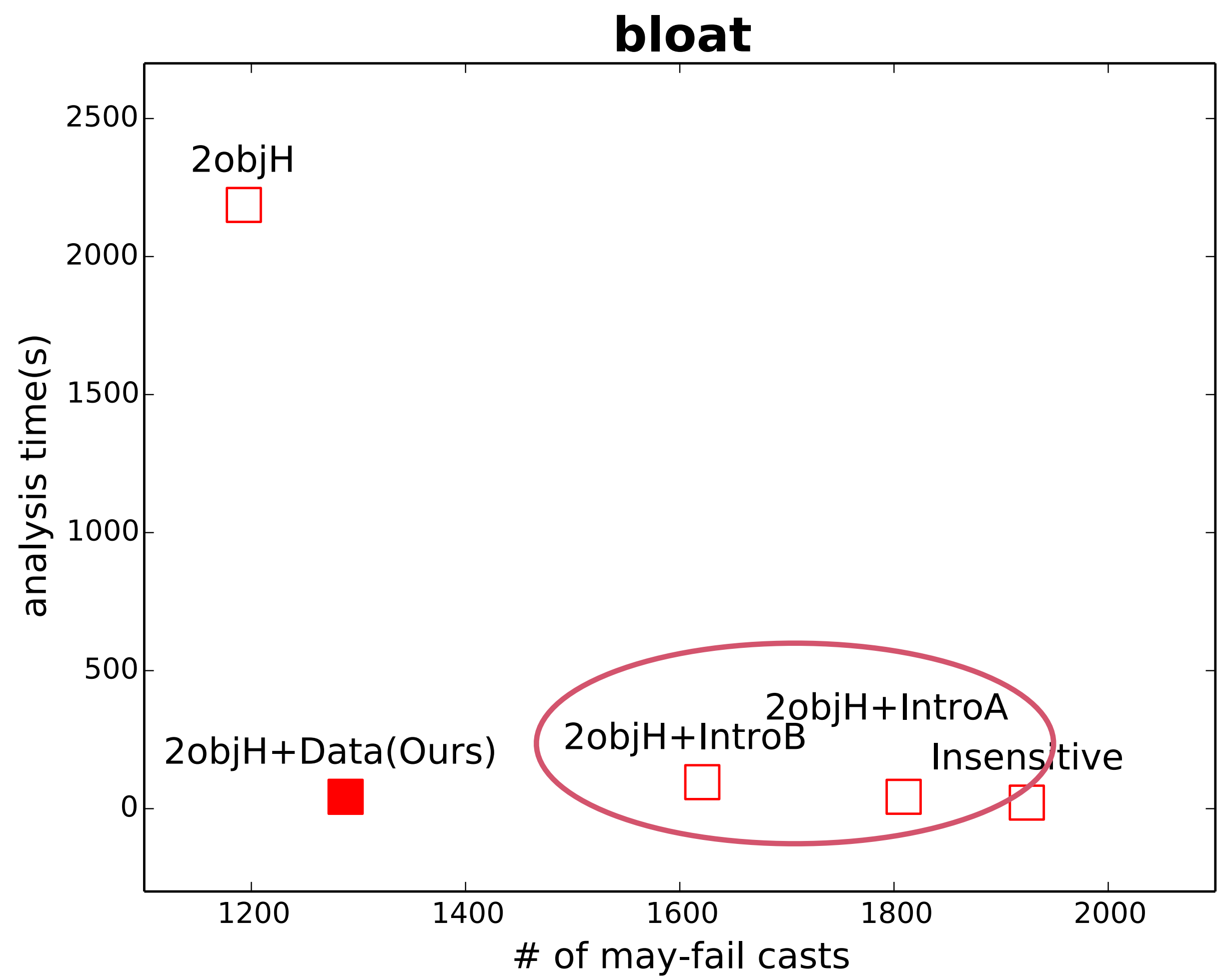


Performance

- Training with 4 small programs from DaCapo, and applied to 6 large programs (1 for validation)

- Machine-tuning outperforms hand-tuning





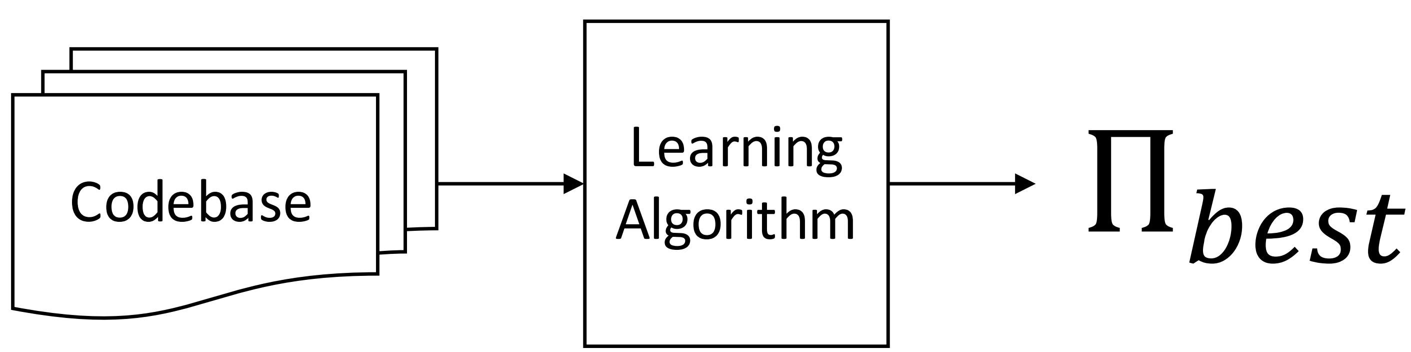
Details

Data-Driven Context Sensitivity

- Parameterized heuristics.

$$H_{\Pi}: M_P \rightarrow \{0,1,\ldots,k\}$$

- Learning heuristics for context-sensitivity



Model: Disjunctive Model

$$
\begin{aligned}
M_1 &: \{a_1, a_2\} \\
M_2 &: \{a_1\} \\
M_3 &: \{a_2\} \\
M_4 &: \{\}
\end{aligned}
$$

$$f_2 = (a_1 \wedge a_2)$$

$$f_1 = a_1 \vee a_2$$

Model: Disjunctive Model

$$M_1 : \{a_1, a_2\}$$
$$M_2 : \{a_1\}$$
$$M_3 : \{a_2\}$$
$$M_4 : \{\}$$

$$f_2 = (a_1 \wedge a_2)$$
$$\{M_1\}$$
$$f_1 = a_1 \vee a_2$$
$$\{M_1, M_2, M_3\}$$

Model: Disjunctive Model

$$\prod = < f_2, f_1 >$$

$$M_1 \rightarrow 2,$$

$$M_2 \rightarrow 1,$$

$$M_3 \rightarrow 1,$$

$$M_4 \rightarrow 0,$$

$$f_2 = (a_1 \wedge a_2)$$

$$\{M_1\}$$

$$f_1 = a_1 \vee a_2$$

$$\{M_1, M_2, M_3\}$$

Atomic Features

“Does the method has a specific word in its signature string?”

Signature features									
#1	“java”	#3	“sun”	#5	“void”	#7	“int”	#9	“String”
#2	“lang”	#4	“()”	#6	“security”	#8	“util”	#10	“init”

Statement features					
#11	AssignStmt	#16	BreakpointStmt	#21	LookupStmt
#12	IdentityStmt	#17	EnterMonitorStmt	#22	NopStmt
#13	InvokeStmt	#18	ExitMonitorStmt	#23	RetStmt
#14	ReturnStmt	#19	GotoStmt	#24	ReturnVoidStmt
#15	ThrowStmt	#20	IfStmt	#25	TableSwitchStmt

“Does the method has a specific type of statement in its body?”

Optimization Problem

- $\Pi = \langle f_1, f_2, \ldots, f_k \rangle$
- Find Π that minimizes

$$\sum_{P \in Pgm} \mathrm{cost}(F_P(H_\Pi(P)))$$

- while satisfying

$$\frac{\sum_{P \in Pgm} |\mathrm{proved}\big(F_P(H_\Pi(P))\big)|}{\sum_{P \in Pgm} |\mathrm{proved}\big(F_P(k)\big)|} \geq \gamma$$

Problem Decomposition Reduces Search Space

Goal

Find Π that minimizes cost while satisfying target precision

Learning Π as a whole

$$\Pi = \langle \boxed{f_1, f_2, \dots, f_{k-1}, f_k} \rangle \Rightarrow |S|^k$$

Set of all possible Boolean formulas

Our approach: Learning one f at a time

$$\Pi = \langle true, \dots, true, \boxed{f_k} \rangle$$

Problem Decomposition Reduces Search Space

Goal

Find Π that minimizes cost while satisfying target precision

Learning Π as a whole

Set of all possible Boolean formulas

$$\Pi = \boxed{\langle f_1, f_2, \ldots, f_{k-1}, f_k \rangle} \Rightarrow |S|^k$$

Our approach: Learning one f at a time

$$\Pi = \langle true, \ldots, true, \boxed{f_{k-1}}, f_k \rangle$$

Problem Decomposition Reduces Search Space

Goal

Find Π that minimizes cost while satisfying target precision

Learning Π as a whole

$$\Pi = \langle \boxed{f_1, f_2, \dots, f_{k-1}, f_k} \rangle \Rightarrow |S|^k$$

Set of all possible Boolean formulas

Our approach: Learning one f at a time

$$\Pi = \langle true, \boxed{f_2}, \dots, f_{k-1}, f_k \rangle$$

Problem Decomposition Reduces Search Space

Goal

Find Π that minimizes cost while satisfying target precision

Learning Π as a whole

Set of all possible Boolean formulas

$$\Pi = \langle \boxed{f_1, f_2, \ldots, f_{k-1}, f_k} \rangle \Rightarrow |S|^k$$

Our approach: Learning one f at a time

$$\Pi = \langle \boxed{f_1}, f_2, \ldots, f_{k-1}, f_k \rangle$$

Problem Decomposition Reduces Search Space

Goal

Find Π that minimizes cost
while satisfying target precision

Learning Π as a whole

Set of all possible Boolean formulas

$$\Pi = \langle \boxed{f_1, f_2, \ldots, f_{k-1}, f_k} \rangle \Rightarrow |S|^k$$

Our approach: Learning one f at a time

$$\Pi = \langle f_1, f_2, \ldots, f_{k-1}, f_k \rangle \Rightarrow k \cdot |S|$$

Has Same Power? Yes.

- We have a theorem for it.
- Please consult with our paper.

Illustration of Algorithm for Searching Boolean Formula f

Initial formula

$a_1 \lor \neg a_1 \lor a_2 \lor \neg a_2$

The most general formula

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

Performance Table

GOAL: Find f that proves all queries

$$W = \{a_1, \neg a_1, a_2, \neg a_2\}$$

Refinement Targets

Iteration 1

$a_1 \vee \neg a_1 \vee a_2 \vee \neg a_2$

	Proven Qs	Cost
$\boldsymbol{a_1}$	Q1, Q3	**20**
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

Pick the most expensive clause

$$W = \{a_1, \neg a_1, a_2, \neg a_2\}$$

Iteration 1

$$a_1 \vee \neg a_1 \vee a_2 \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	**20**
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

$\boxed{\text{Remove the clause from the workset}}$

$$W = \{\cancel{a_1}, \neg a_1, a_2, \neg a_2\}$$

Iteration 1

$(a_1 \wedge a_2) \vee \neg a_1 \vee a_2 \vee \neg a_2$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

Refine the clause conservatively

$$W = \{\neg a_1, a_2, \neg a_2\}$$

Iteration 1

$$(a_1 \wedge a_2) \vee \neg a_1 \vee a_2 \vee \neg a_2$$

$$\frac{\sum_{P \in Pgm} |\text{proved}\big(F_P(H_\Pi(P))\big)|}{\sum_{P \in Pgm} |\text{proved}\big(F_P(k)\big)|} = 1$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

Check whether the formula satisfies the precision goal

$$W = \{\neg a_1, a_2, \neg a_2\}$$

Iteration 1

$$(a_1 \wedge a_2) \vee \neg a_1 \vee a_2 \vee \neg a_2$$

$$F_P\big(H_\Pi(P)\big): \{Q1, Q2, Q3, Q4\}$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

Current formula proves all queries

$$W = \{\neg a_1, a_2, \neg a_2\}$$

Iteration 1

subset

$\subset$

$$(\cancel{a_1} \lor a_2) \lor \neg a_1 \lor a_2 \lor \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

Drop the refined clause from the formula

$$W = \{\neg a_1, a_2, \neg a_2\}$$

Iteration 2

$$\neg a_1 \vee a_2 \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

$$W = \{a_1, \neg a_1, a_2\}$$

Iteration 2

$$\neg a_1 \vee a_2 \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	**15**
$\neg a_2$	Q1, Q2, Q4	10

Again, pick the most expensive clause

$$W = \{a_1, \neg a_1, a_2\}$$

Iteration 2

$$\neg a_1 \lor a_2 \lor \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
$\boldsymbol{a_2}$	Q1, Q2, Q3	**15**
$\neg a_2$	Q1, Q2, Q4	10

Remove the clause from the workset

$$W = \{a_1, \neg a_1, \cancel{a_2}\}$$

Iteration 2

$$\neg a_1 \vee (a_1 \wedge a_2) \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

$\boxed{\text{Refine the clause conservatively}}$

$$W = \{a_1, \neg a_1\}$$

Iteration 2

$$\neg a_1 \vee (a_1 \wedge a_2) \vee \neg a_2$$

$$\frac{\sum_{P \in Pgm} |\mathrm{proved}\left(F_P(H_\Pi(P))\right)|}{\sum_{P \in Pgm} |\mathrm{proved}\left(F_P(k)\right)|} = 1$$

	Proven Qs	**Cost**
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

Check whether the formula satisfies the precision goal

$$W = \{a_1, \neg a_1\}$$

Iteration 2

$$\neg a_1 \vee (a_1 \wedge a_2) \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10

$$F_P\big(H_\Pi(P)\big): \{Q1, Q2, Q3, Q4\}$$

Still, the formula proves all queries

$$W = \{a_1, \neg a_1\}$$

Iteration 2

$$\neg a_1 \vee (a_1 \wedge a_2) \vee \neg a_2$$

	Proven Qs	Cost
a_1	**Q1, Q3**	20
$\neg a_1$	Q2	7
a_2	**Q1, Q2, Q3**	**15**
$\neg a_2$	Q1, Q2, Q4	10

Refined clause passes subset checking

$$W = \{a_1, \neg a_1\}$$

Iteration 2

$$\neg a_1 \vee (\overset{\color{red}{C_1}}{a_1} \wedge a_2) \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	**Q1, Q2, Q4**	**13**

$\boxed{\text{Record the refined clause } C_1}$

$$W = \{a_1, \neg a_1\}$$

Iteration 2

$$\neg a_1 \vee (\overset{\boldsymbol{C_1}}{a_1} \wedge a_2) \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10
$\boldsymbol{C_1}$	**Q1, Q2, Q4**	**13**

...and add C_1 to the workset for further refinement

$$W = \{a_1, \neg a_1, (\boldsymbol{a_1} \wedge \boldsymbol{a_2})\}$$

Iteration 3

$$\underbrace{}^{C_1}$$

$$\neg a_1 \lor (a_1 \land a_2) \lor \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	13

$$W = \{a_1, \neg a_1, (a_1 \land a_2)\}$$

Iteration 3

C_1

$$\neg a_1 \vee (a_1 \wedge a_2) \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	**13**

Pick the most expensive clause

$$W = \{a_1, \neg a_1, (a_1 \wedge a_2)\}$$

Iteration 3

$$\neg a_1 \vee (\overbrace{a_1 \wedge a_2}^{C_1}) \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	**13**

Remove the clause from the workset

$$W = \{a_1, \neg a_1, \cancel{(a_1 \wedge a_2)}\}$$

Iteration 3

$$\overset{C_1}{\neg a_1 \vee (a_1 \wedge a_2 \wedge \neg a_2) \vee \neg a_2}$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	**Q1, Q2, Q4**	10
C_1	**Q1, Q2, Q4**	**13**

Refine the clause conservatively

$$W = \{a_1, \neg a_1\}$$

Iteration 3

$$\neg a_1 \vee (\overbrace{a_1 \wedge a_2}^{C_1} \wedge \neg a_2) \vee \neg a_2$$

$$\frac{\sum_{P \in Pgm} |\text{proved}\big(F_P(H_\Pi(P))\big)|}{\sum_{P \in Pgm} |\text{proved}\big(F_P(k)\big)|} = 1$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	**Q1, Q2, Q4**	10
C_1	**Q1, Q2, Q4**	**13**

Check whether the formula satisfies the precision goal

$$W = \{a_1, \neg a_1\}$$

Iteration 3

$$\neg a_1 \vee (\overset{C_1}{a_1} \wedge a_2 \wedge \neg a_2) \vee \neg a_2$$

$$F_P\big(H_\Pi(P)\big): \{Q1, Q2, \cancel{Q3}, Q4\}$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	**Q1, Q2, Q4**	10
C_1	**Q1, Q2, Q4**	**13**

This refinement failed to prove all queries

$$W = \{a_1, \neg a_1\}$$

Iteration 3

$$\neg a_1 \vee (\overset{C_1}{a_1} \wedge a_2 \wedge \cancel{\neg a_2}) \vee \neg a_2$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	**Q1, Q2, Q4**	10
C_1	**Q1, Q2, Q4**	**13**

$\boxed{\text{Revert to the last state}}$

$$W = \{a_1, \neg a_1\}$$

After iteration 4 and 5...

$$\underbrace{(a_1 \wedge a_2)}_{C_1} \vee \underbrace{(\neg a_2 \wedge \neg a_1)}_{C_2}$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	13
C_2	Q1, Q2, Q3	

$$W = \{(\neg a_2 \wedge \neg a_1)\}$$

Iteration 6

$$\overset{C_1}{(a_1 \wedge a_2)} \vee \overset{C_2}{(\neg a_2 \wedge \neg a_1)}$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	13
$\boldsymbol{C_2}$	Q1, Q2, Q3	**7**

Remove the clause from the workset

$$W = \{\cancel{(\neg \boldsymbol{a_2} \wedge \neg \boldsymbol{a_1})}\}$$

Iteration 6

$$\overset{C_1}{(a_1 \wedge a_2)} \vee \overset{C_2}{(\neg a_2 \wedge \neg a_1 \wedge a_2)}$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
$\boldsymbol{a_2}$	**Q1, Q2, Q3**	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	13
$\boldsymbol{C_2}$	**Q1, Q2, Q3**	**7**

Refine the clause conservatively

$$W = \emptyset$$

Iteration 6

$$\underbrace{(a_1 \wedge a_2)}_{C_1} \vee \underbrace{(\neg a_2 \wedge \neg a_1 \wedge a_2)}_{C_2}$$

$$\frac{\sum_{P \in Pgm} |\mathrm{proved}(F_P(H_\Pi(P)))|}{\sum_{P \in Pgm} |\mathrm{proved}(F_P(k))|} = 1$$

	Proven Qs	**Cost**
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	**Q1, Q2, Q3**	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	13
C_2	**Q1, Q2, Q3**	**7**

Check whether the formula satisfies the precision goal

$$W = \emptyset$$

Iteration 6

$$\overset{C_1}{(a_1 \wedge a_2)} \vee \overset{C_2}{(\neg a_2 \wedge \neg a_1 \wedge a_2)}$$

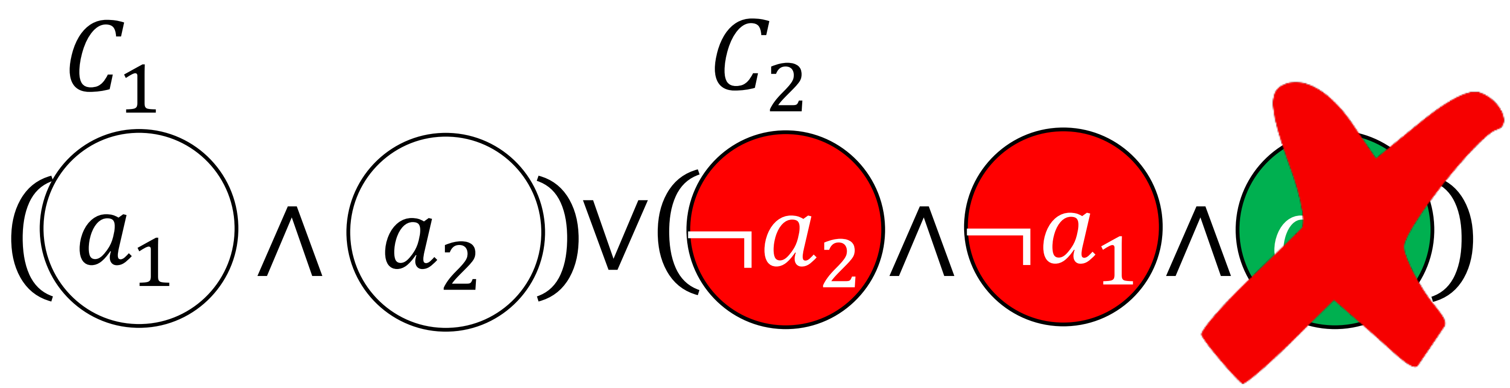
$$F_P\big(H_\Pi(P)\big): \{Q1, Q2, \cancel{Q3}, Q4\}$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	**Q1, Q2, Q3**	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	13
C_2	**Q1, Q2, Q3**	**7**

$$\boxed{\text{This refinement failed to prove all queries}}$$

$$W = \emptyset$$

Iteration 6



	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	**Q1, Q2, Q3**	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	13
C_2	**Q1, Q2, Q3**	**7**

Revert to the last state

$W = \emptyset$

END

$$\overset{C_1}{(a_1 \wedge a_2)} \vee \overset{C_2}{(\neg a_2 \wedge \neg a_1)}$$

	Proven Qs	Cost
a_1	Q1, Q3	20
$\neg a_1$	Q2	7
a_2	Q1, Q2, Q3	15
$\neg a_2$	Q1, Q2, Q4	10
C_1	Q1, Q2, Q4	13
C_2	Q1, Q2, Q3	7

Algorithm ends when there is no refineable claus.

$$\boldsymbol{W} = \emptyset$$

Experiments

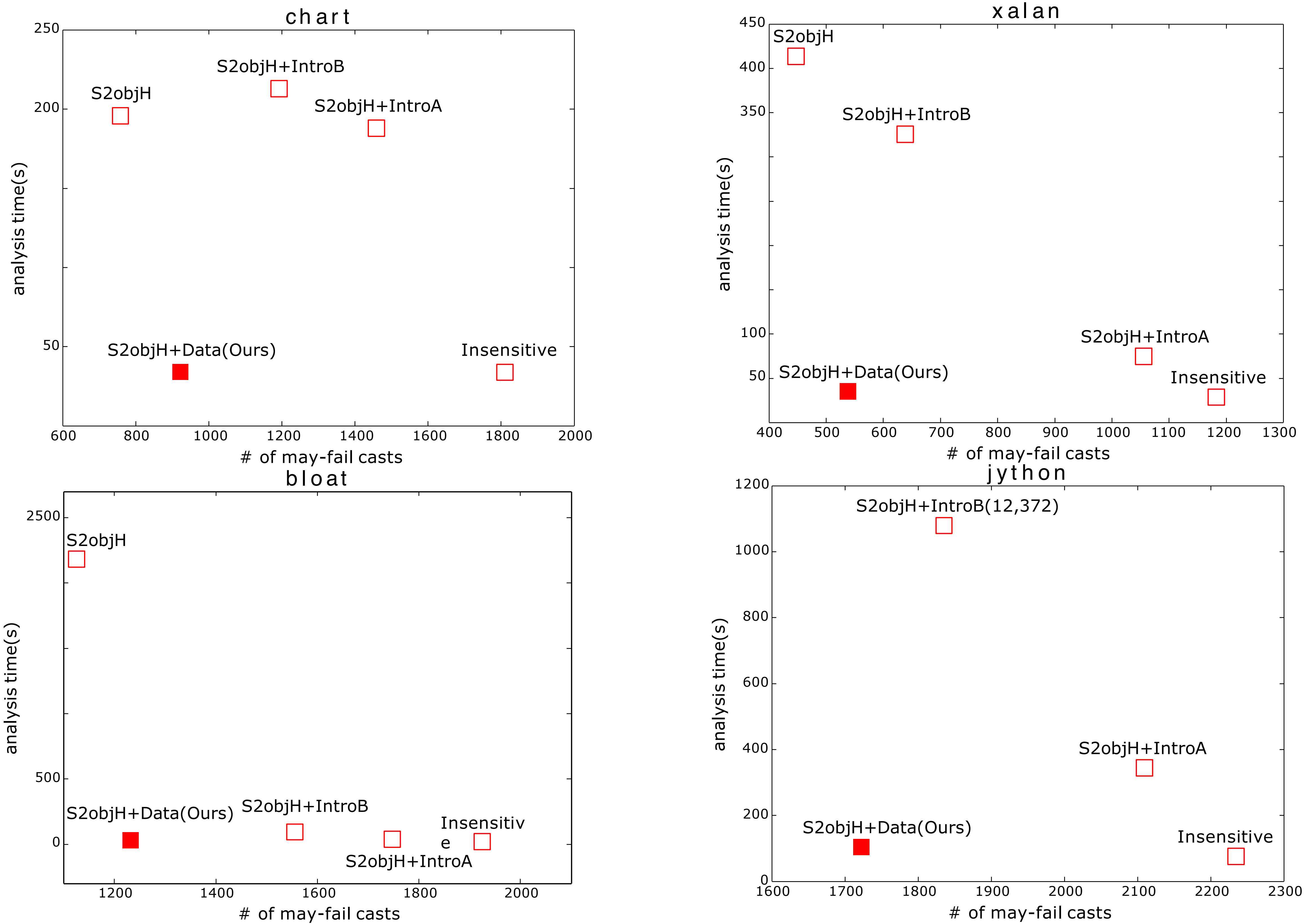
Setting

- DaCapo Benchmark suite
  - Four small programs for training
  - One large programs for validation
  - Five large programs for testing

Research Questions

- RQ1: Effectiveness
- RQ2: Model adequacy

RQ1: Effectiveness



RQ2: vs. Linear Model

- On same time budget for training, disjunctive model out performs linear model

Benchmarks	Non-disjunctive		Disjunctive(Ours)	
	may-fail casts	time(s)	may-fail casts	time(s)
eclipse	946	25	596	21
chart	1,569	48	937	33
bloat	1,771	46	1,232	27
xalan	996	42	539	33
jython	2069	346	1,738	104
Total	7,352	346	5,042	218

Conclusion

Win

Selective Context Sensitivity