Testing the Binding Code of Scripting Languages with Cooperative Mutation

Peng Xu Yanhao Wang **Hong Hu** Purui Su

PennState

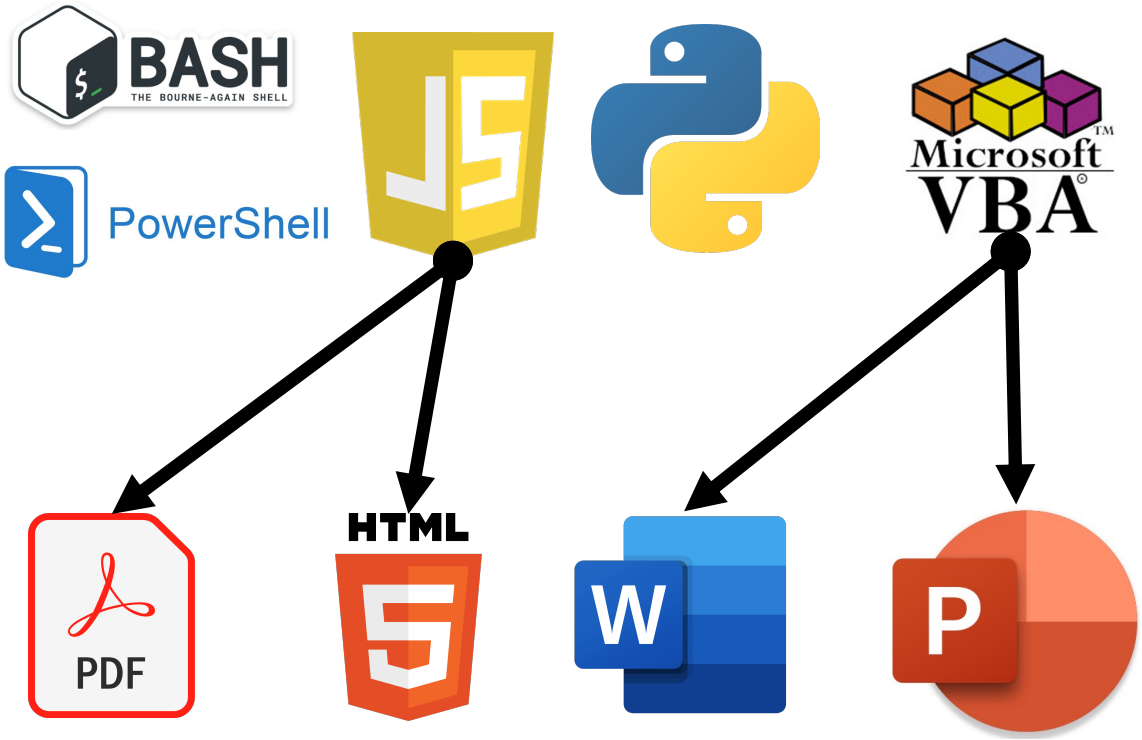
Scripting Language

Scripting Language

Scripting Language integrated to documents

Scripting Language integrated to documents

Scripting Language integrated to documents

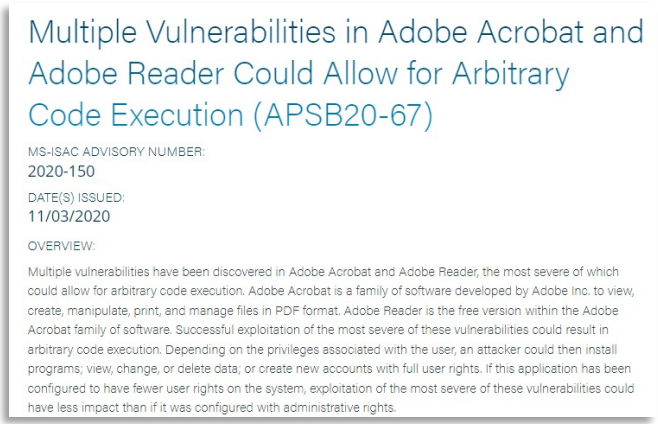


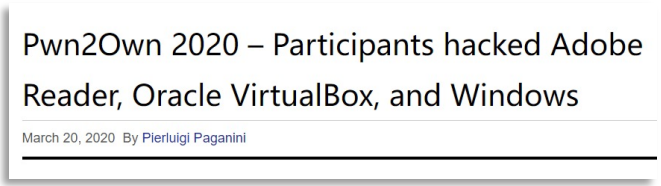
Vulnerability in Embedded Scripting Language

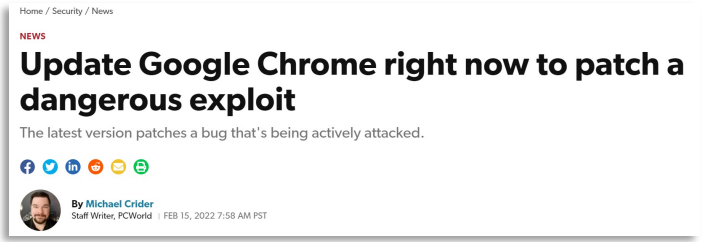
- Dangerous and Common

Vulnerability in Embedded Scripting Language

- Dangerous and Common







Our work: Cooper

Our work: Cooper

- Cooperative mutation
  - effectively test binding code of scripting languages

Our work: Cooper

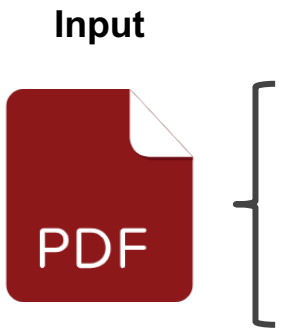
- Cooperative mutation
  - effectively test binding code of scripting languages
- 134 bugs in Adobe Acrobat, Foxit Reader, and Microsoft Word
  - 33 CVE and 22K dollars bounty

Our work: Cooper

- Cooperative mutation
  - effectively test binding code of scripting languages
- 134 bugs in Adobe Acrobat, Foxit Reader, and Microsoft Word
  - 33 CVE and 22K dollars bounty
- Open-sourced at: https://github.com/TCA-ISCAS/Cooper

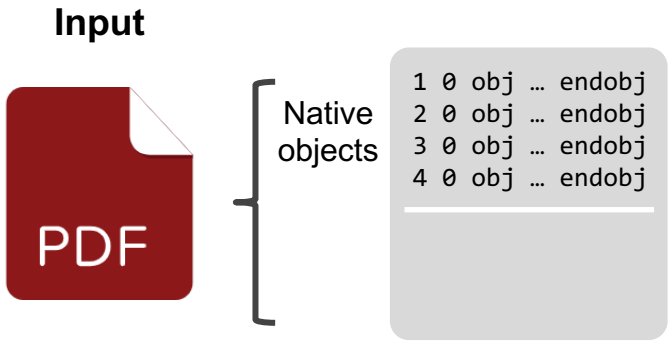
Document processing programs

- Input: Native objects + Scripts code



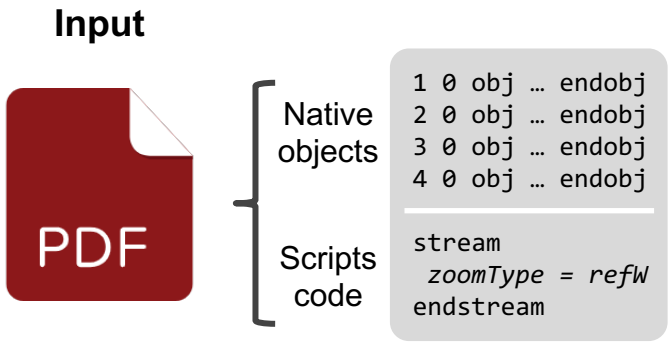
Document processing programs

- Input: Native objects + Scripts code



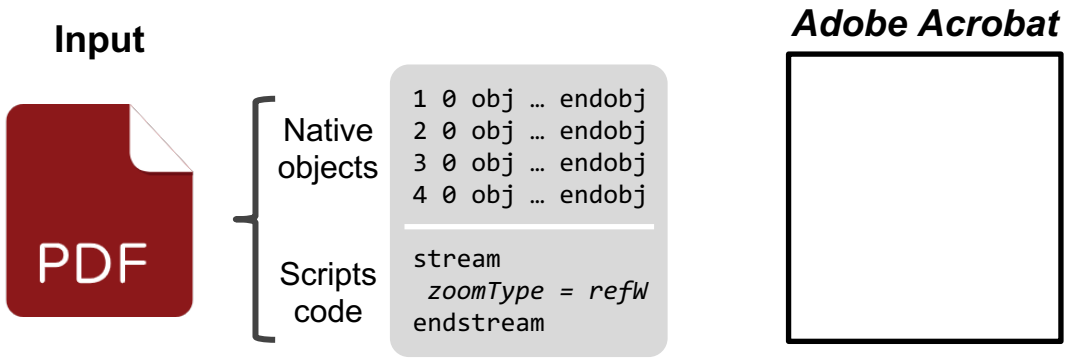
Document processing programs

- Input: Native objects + Scripts code



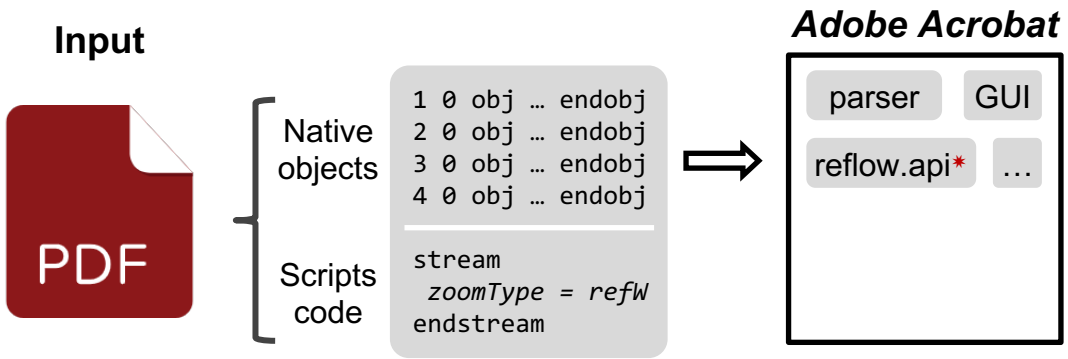
Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs



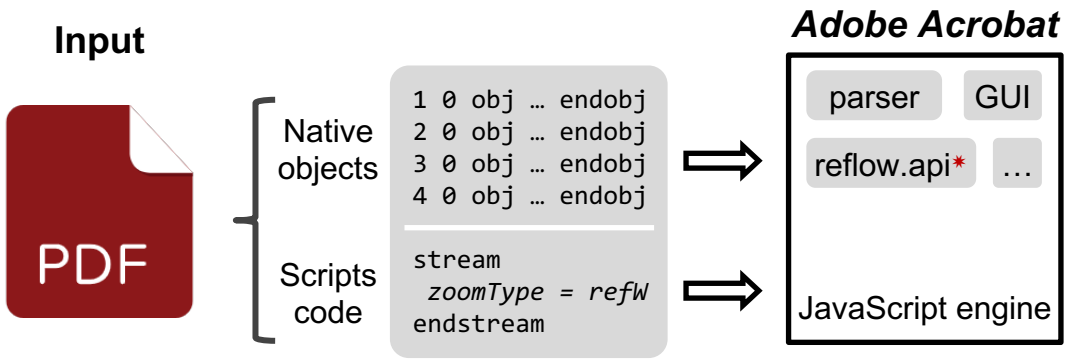
Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs



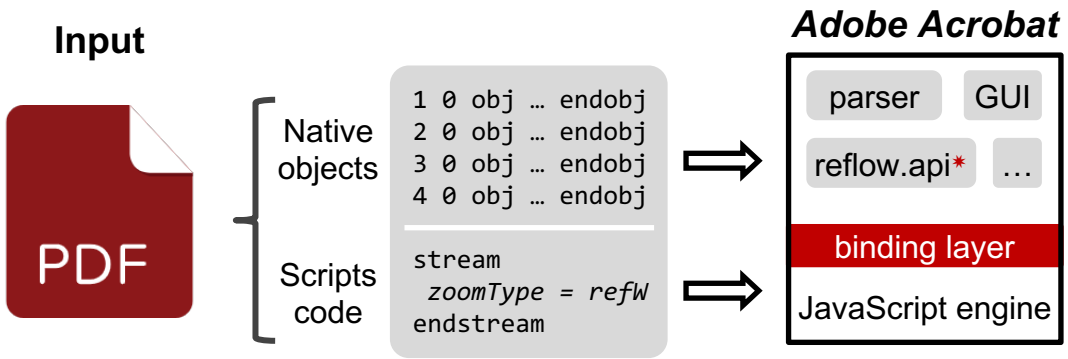
Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs



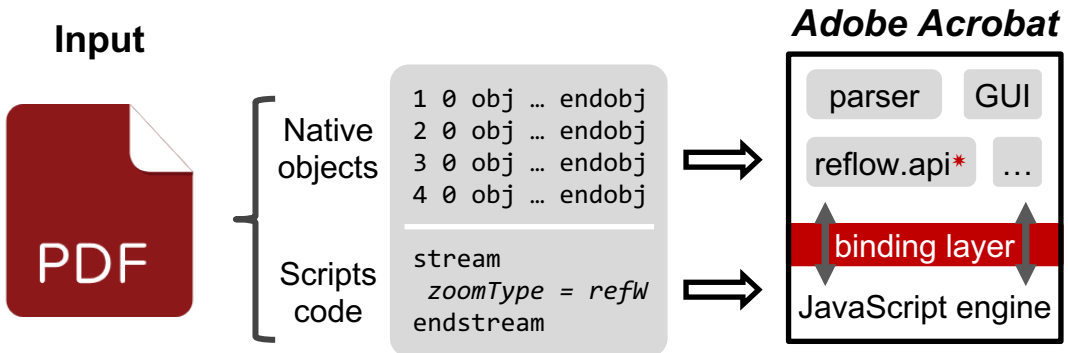
Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs
- Binding layer connects two components



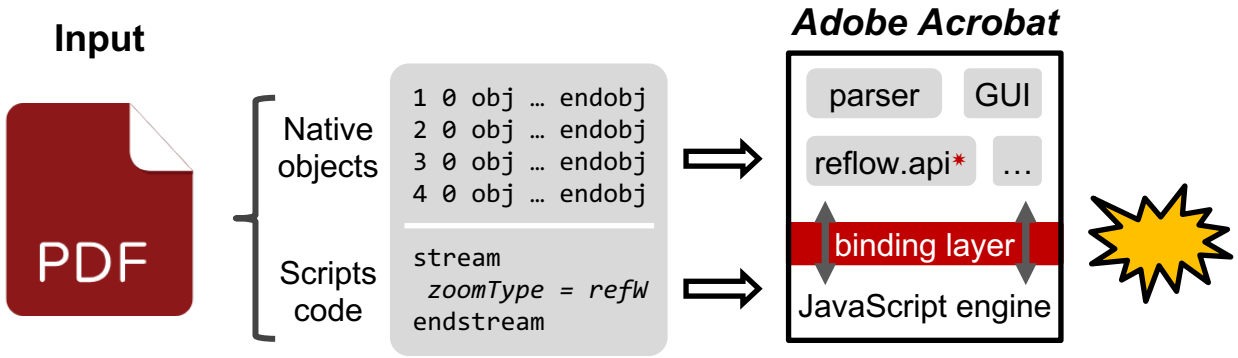
Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs
- Binding layer connects two components



Document processing programs

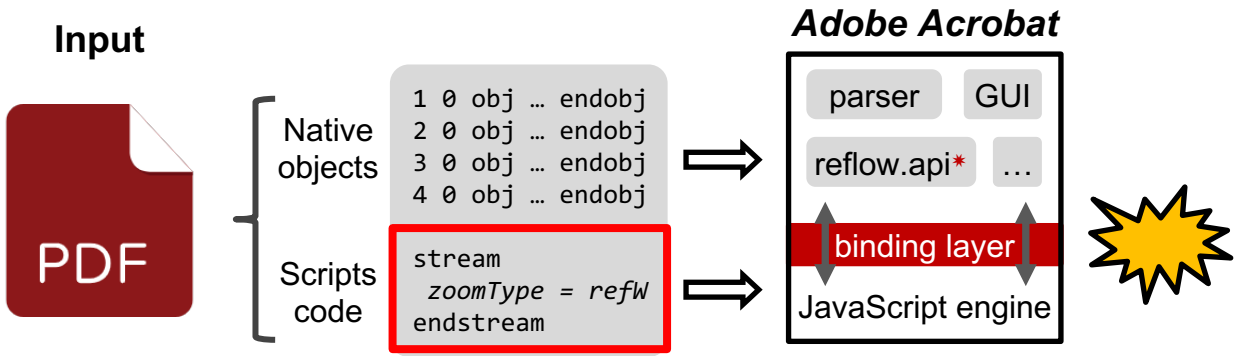
- Input: Native objects + Scripts code
- Two components for processing inputs
- Binding layer connects two components



- Binding layer is too complicated, leading to BUGS

Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs
- Binding layer connects two components



Previous work only mutates scripts code

- Binding layer is too complicated, leading to BUGS

Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- $2.5K bounty

```
%PDF-1.3
1 0 obj << /Pages 2 0 R >> endobj
2 0 obj << /Kids [ 3 0 R ] >> endobj
3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
          /AA << /O << /S /JavaScript
                       /JS 5 0 R >> >> >> endobj
4 0 obj << /FirstChar 0
          /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
          /LastChar 255 >> endobj
5 0 obj << /Length 539 >>
        stream
          this.zoomType=zoomtype.refW;    % Trigger the bug
        endstream
    endobj
trailer << /Root 1 0 R >>
```

Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- $2.5K bounty

How to trigger this vulnerability?

```
%PDF-1.3
1 0 obj << /Pages 2 0 R >> endobj
2 0 obj << /Kids [ 3 0 R ] >> endobj
3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
          /AA << /O << /S /JavaScript
                       /JS 5 0 R >> >> >> endobj
4 0 obj << /FirstChar 0
          /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
          /LastChar 255 >> endobj
5 0 obj << /Length 539 >>
        stream
          this.zoomType=zoomtype.refW;    % Trigger the bug
        endstream
    endobj
trailer << /Root 1 0 R >>
```

Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- $2.5K bounty

How to trigger this vulnerability?

- Native Objects:

Insert an extra element into Font's Widths array.

```
1  %PDF-1.3
2  1 0 obj << /Pages 2 0 R >> endobj
3  2 0 obj << /Kids [ 3 0 R ] >> endobj
4  3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
5             /AA << /O << /S /JavaScript
6                         /JS 5 0 R >> >> >> endobj
7  4 0 obj << /FirstChar 0
8             /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
9             /LastChar 255 >> endobj
10 5 0 obj << /Length 539 >>
11          stream
12             this.zoomType=zoomtype.refW;    % Trigger the bug
13          endstream
14      endobj
15 trailer << /Root 1 0 R >>
```

Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- $2.5K bounty

How to trigger this vulnerability?

- Native Objects:

Insert an extra element into Font's Widths array.

- Scripts Code:

Invoke this.zoomType=zoomtype.refW;

```
1  %PDF-1.3
2  1 0 obj << /Pages 2 0 R >> endobj
3  2 0 obj << /Kids [ 3 0 R ] >> endobj
4  3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
5             /AA << /O << /S /JavaScript
6                         /JS 5 0 R >> >> >> endobj
7  4 0 obj << /FirstChar 0
8             /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
9             /LastChar 255 >> endobj
10 5 0 obj << /Length 539 >>
11          stream
12             this.zoomType=zoomtype.refW;     % Trigger the bug
13          endstream
14       endobj
15 trailer << /Root 1 0 R >>
```

Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- $2.5K bounty

How to trigger this vulnerability?

- Native Objects:

Insert an extra element into Font's Widths array.

- Scripts Code:

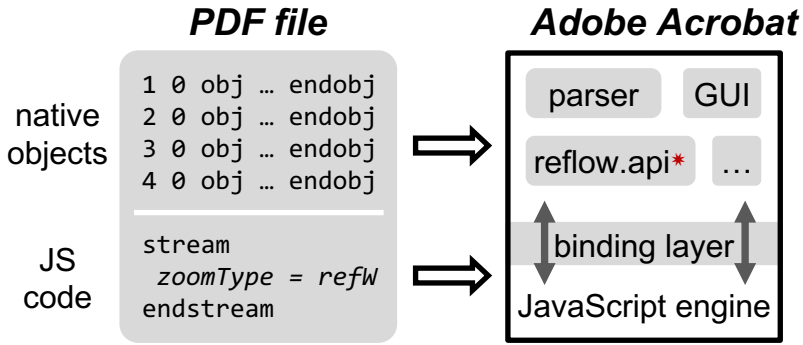
Invoke this.zoomType=zoomtype.refW;

```
1  %PDF-1.3
2  1 0 obj << /Pages 2 0 R >> endobj
3  2 0 obj << /Kids [ 3 0 R ] >> endobj
4  3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
5             /AA << /O << /S /JavaScript
6                         /JS 5 0 R >> >> >> endobj
7  4 0 obj << /FirstChar 0
8             /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
9             /LastChar 255 >> endobj
10 5 0 obj << /Length 539 >>
11          stream
12             this.zoomType=zoomtype.refW;     % Trigger the bug
13          endstream
14      endobj
15 trailer << /Root 1 0 R >>
```

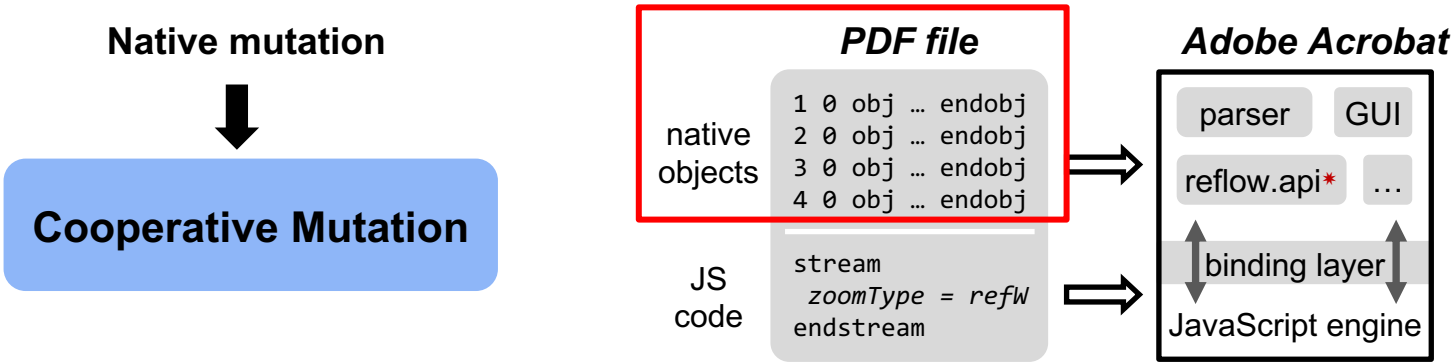
One-dimension-mutation cannot trigger this vulnerability

Our Solution: Cooperative Mutation

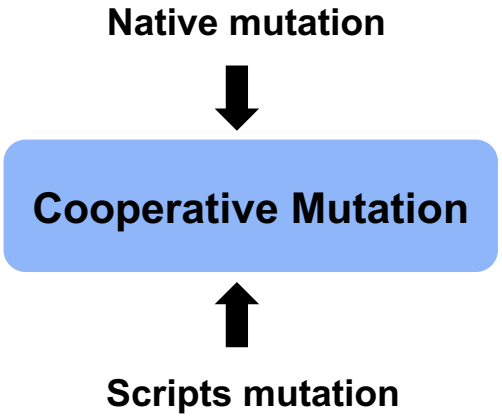
Cooperative Mutation

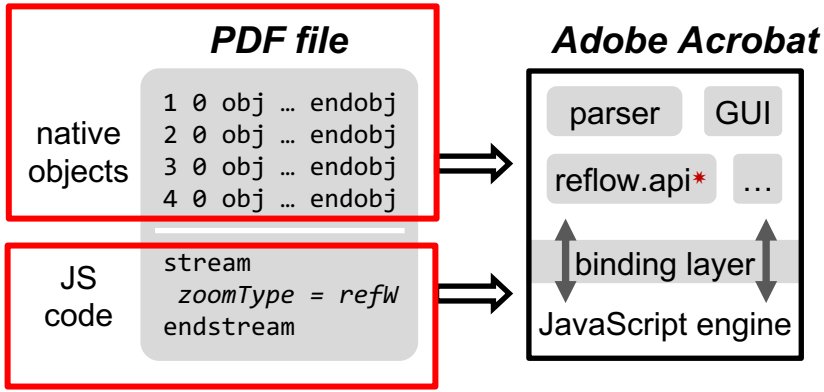


Our Solution: Cooperative Mutation



Our Solution: Cooperative Mutation

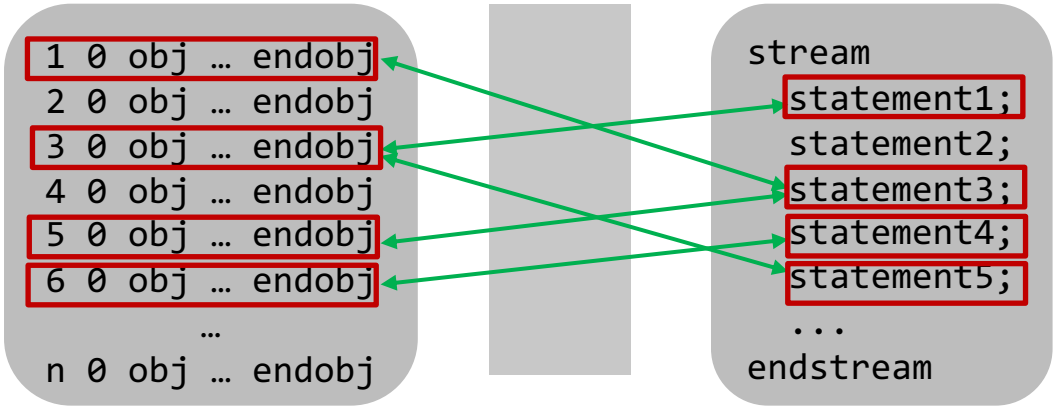




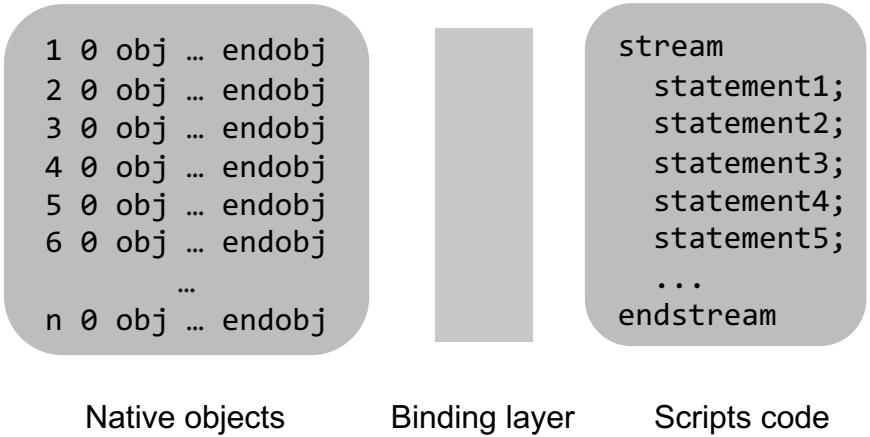
How to mutate objects & code ***cooperatively***?

How to mutate objects & code ***cooperatively***?

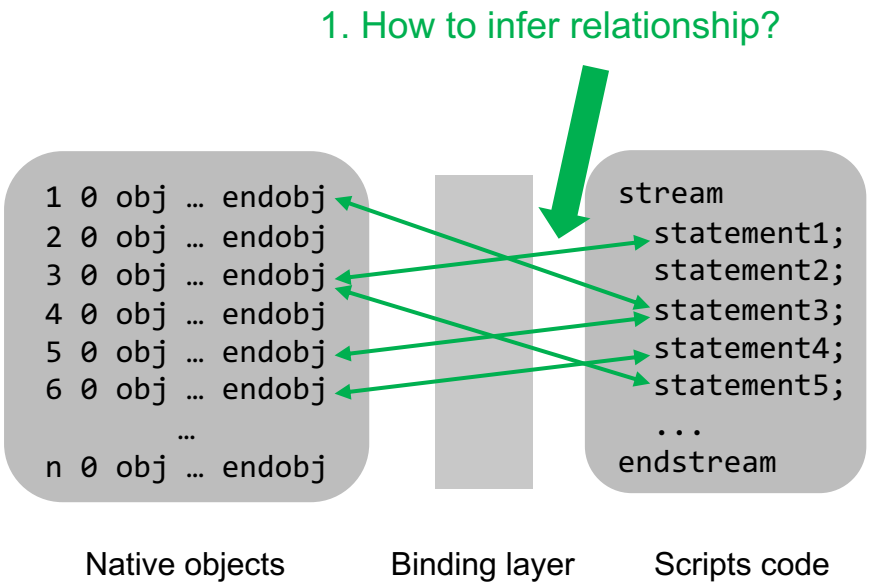
How to mutate objects & code ***cooperatively***?



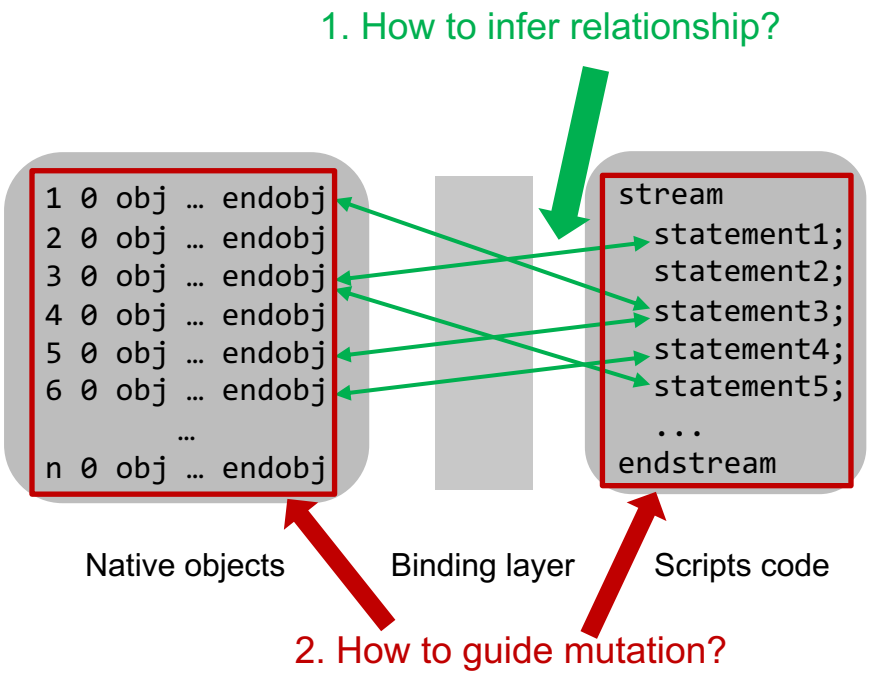
Challenges


1 0 obj … endobj
2 0 obj … endobj
3 0 obj … endobj
4 0 obj … endobj
5 0 obj … endobj
6 0 obj … endobj
…
n 0 obj … endobj
stream
statement1;
statement2;
statement3;
statement4;
statement5;
...
endstream
Native objects
Binding layer
Scripts code

Challenges

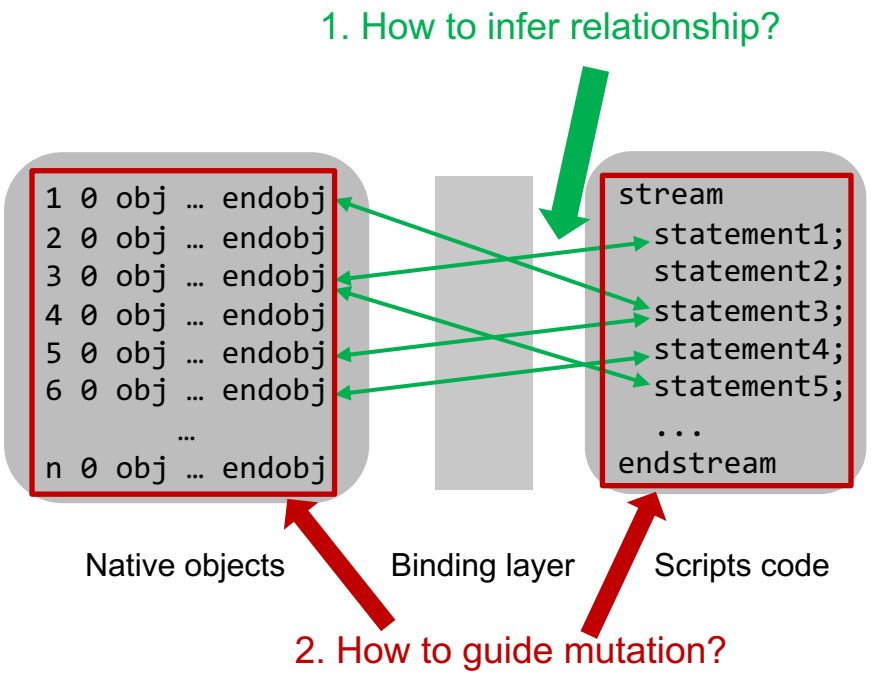

1. How to infer relationship?
1 0 obj … endobj
2 0 obj … endobj
3 0 obj … endobj
4 0 obj … endobj
5 0 obj … endobj
6 0 obj … endobj
…
n 0 obj … endobj
stream
statement1;
statement2;
statement3;
statement4;
statement5;
...
endstream
Native objects
Binding layer
Scripts code

Challenges


1. How to infer relationship?
1 0 obj … endobj
2 0 obj … endobj
3 0 obj … endobj
4 0 obj … endobj
5 0 obj … endobj
6 0 obj … endobj
…
n 0 obj … endobj
stream
statement1;
statement2;
statement3;
statement4;
statement5;
...
endstream
Native objects
Binding layer
Scripts code
2. How to guide mutation?

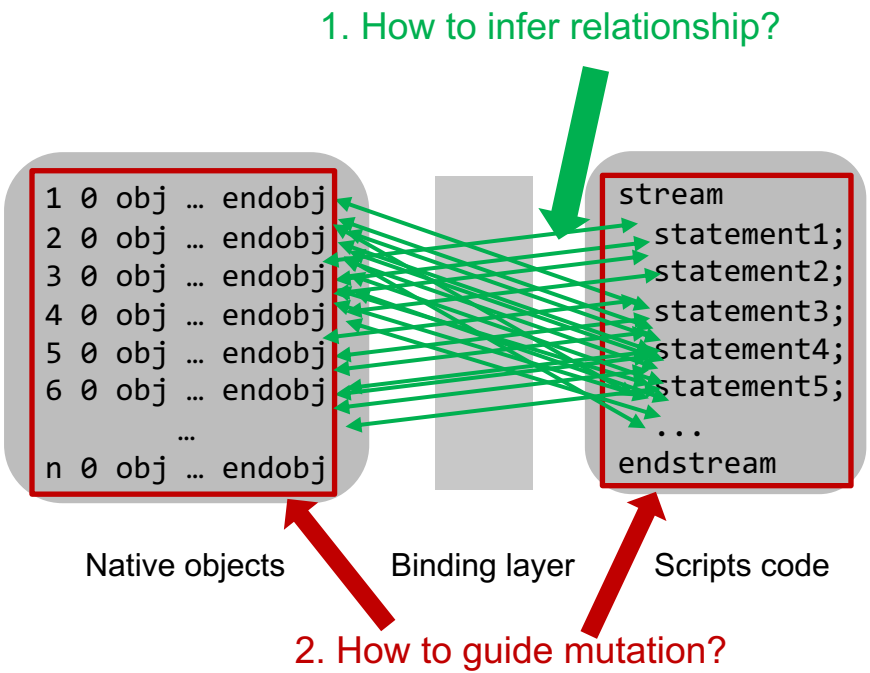
Challenges

Too many objects!!! Makes it hard for inferring and mutation



Challenges

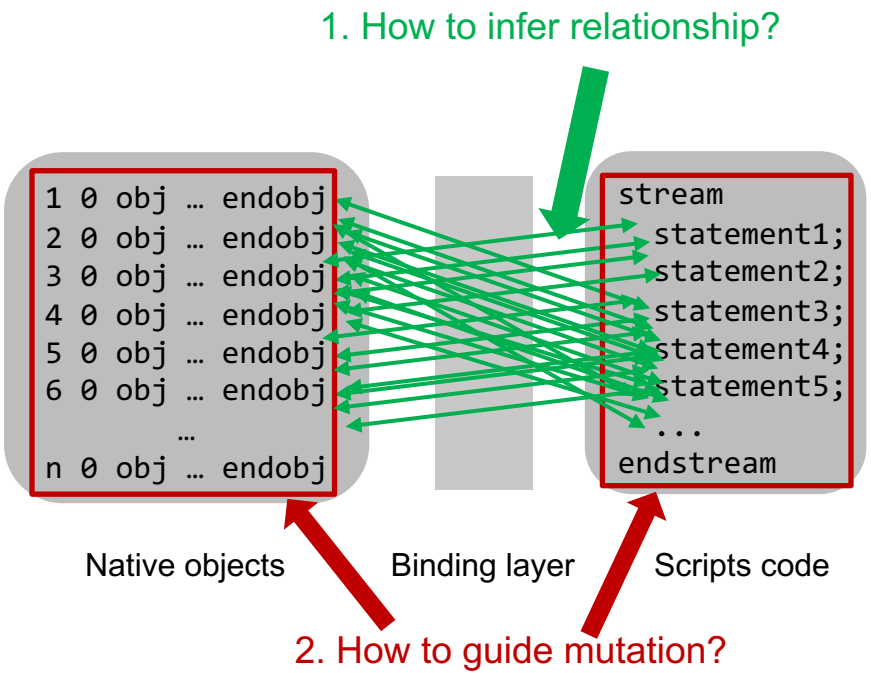
Too many objects!!! Makes it hard for inferring and mutation



Challenges

Too many objects!!! Makes it hard for inferring and mutation

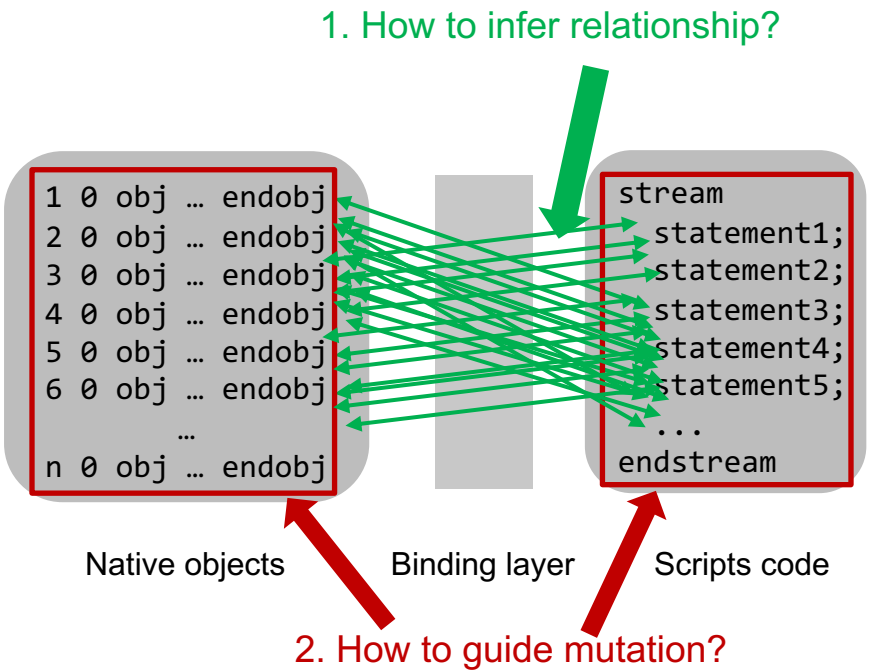
We need to reduce the search space of native objects.



Challenges

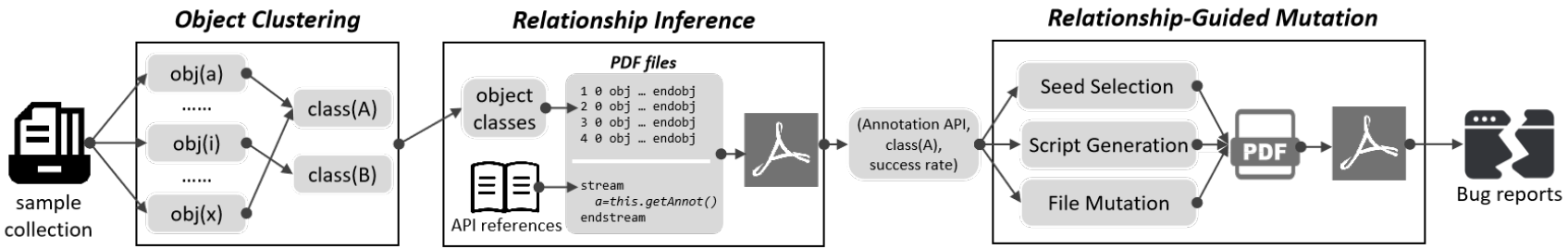
Too many objects!!! Makes it hard for inferring and mutation

We need to reduce the search space of native objects.

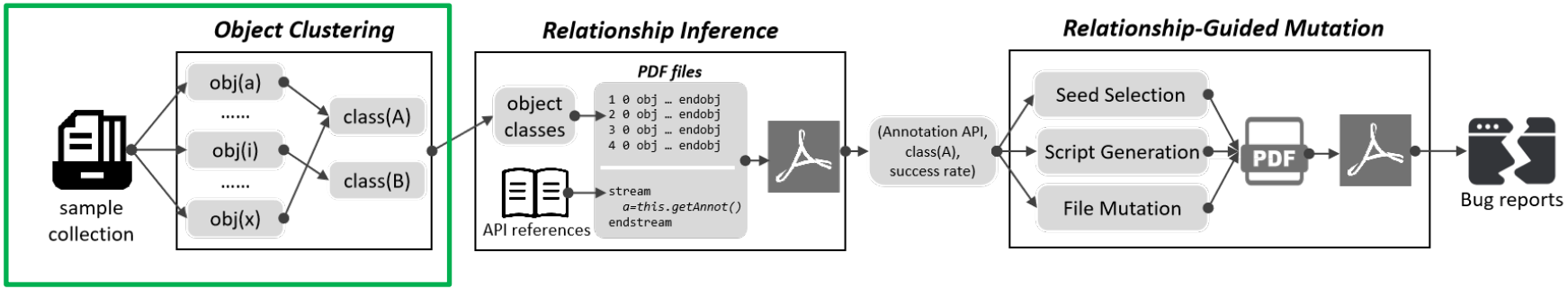


0. How to cluster native objects?

Cooper Overview

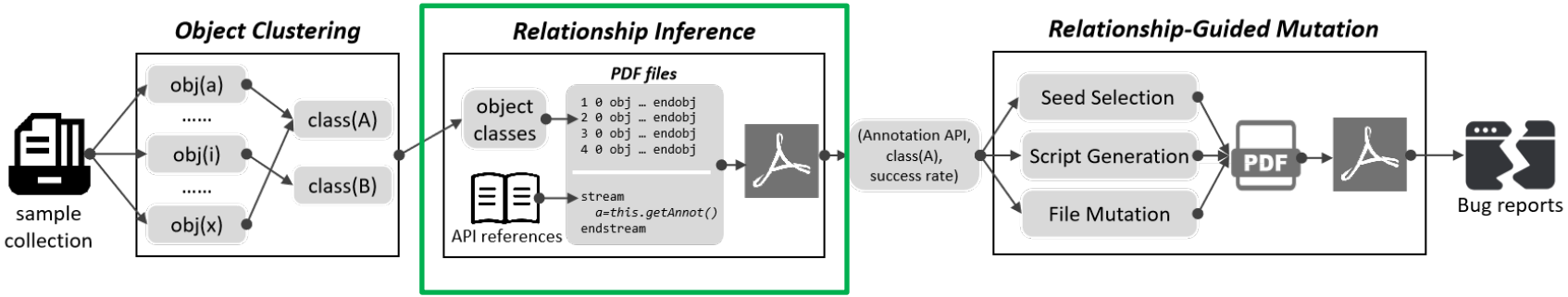


Cooper Overview



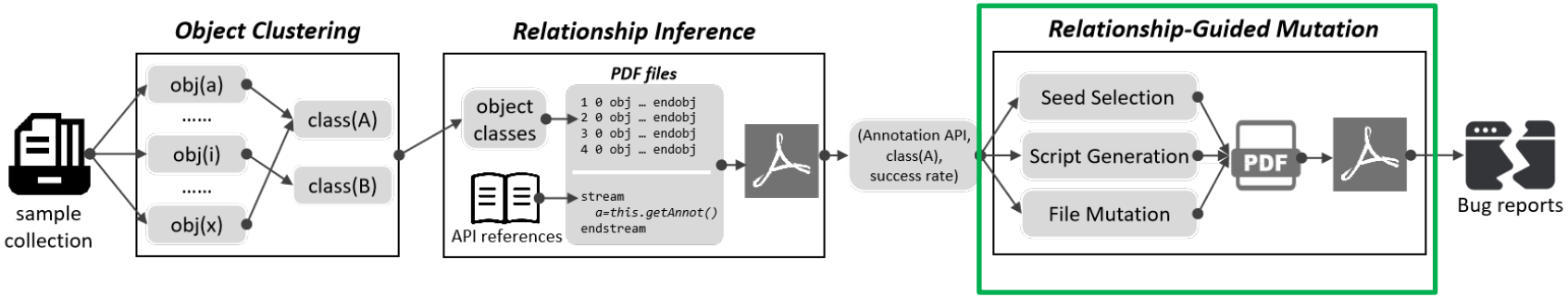
cluster input objects to semantic-similar classes

Cooper Overview



infer relation between native input and script code

Cooper Overview



use the inferred relation
to guide mutation

Object Clustering

Object Clustering

$$O\text{:object} = \left\{ \begin{array}{l} A_0\text{: } \mathbf{name_0} = \text{object}_0, \\ A_1\text{: } \mathbf{name_1} = \text{object}_1, \\ A_2\text{: } \mathbf{name_2} = \text{object}_2, \\ \quad \ldots = \ldots \end{array} \right\}$$

Object Clustering

$$O\text{:object} = \left\{ \begin{array}{c} A_0\text{: } \mathbf{name_0} = \text{object}_0, \\ A_1\text{: } \mathbf{name_1} = \text{object}_1, \\ A_2\text{: } \mathbf{name_2} = \text{object}_2, \\ \ldots = \ldots \end{array} \right\}$$

Name contains semantic information

Object Clustering

$$O\text{:object} = \left\{ \begin{array}{c} A_0\text{: } \mathbf{name_0} = \text{object}_0, \\ A_1\text{: } \mathbf{name_1} = \text{object}_1, \\ A_2\text{: } \mathbf{name_2} = \text{object}_2, \\ \ldots = \ldots \end{array} \right\}$$

Name contains semantic information

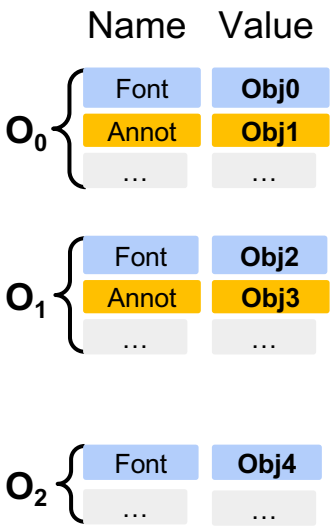
- Clustering objects with **name**

Object Clustering

$$O\text{:object} = \left\{ \begin{array}{l} A_0\text{: } \mathbf{name_0} = \text{object}_0, \\ A_1\text{: } \mathbf{name_1} = \text{object}_1, \\ A_2\text{: } \mathbf{name_2} = \text{object}_2, \\ \quad \ldots = \ldots \end{array} \right\}$$

Name contains semantic information

- Clustering objects with **name**

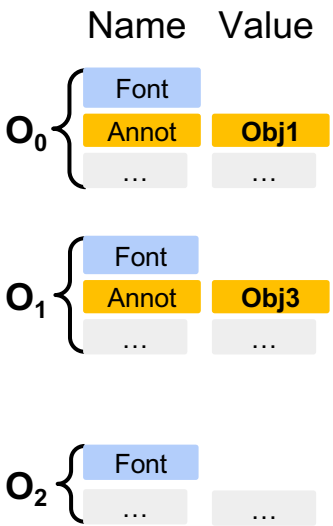


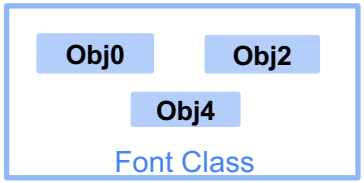
Object Clustering

$$O\text{:object} = \begin{cases} A_0\text{: } \mathbf{name_0} = \text{object}_0, \\ A_1\text{: } \mathbf{name_1} = \text{object}_1, \\ A_2\text{: } \mathbf{name_2} = \text{object}_2, \\ \dots = \dots \end{cases}$$

Name contains semantic information

- Clustering objects with **name**



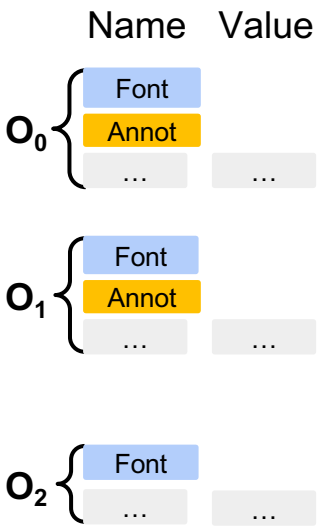


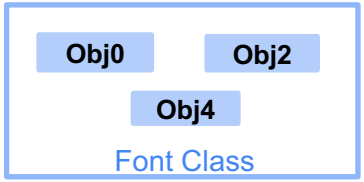
Object Clustering

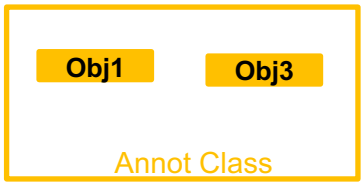
$$O\text{:object} = \left\{ \begin{array}{c} A_0\text{: } \mathbf{name_0} = \text{object}_0, \\ A_1\text{: } \mathbf{name_1} = \text{object}_1, \\ A_2\text{: } \mathbf{name_2} = \text{object}_2, \\ \ldots = \ldots \end{array} \right\}$$

Name contains semantic information

- Clustering objects with **name**







Object Clustering

- Spliting and merging classes with attribute similarity

Object Clustering

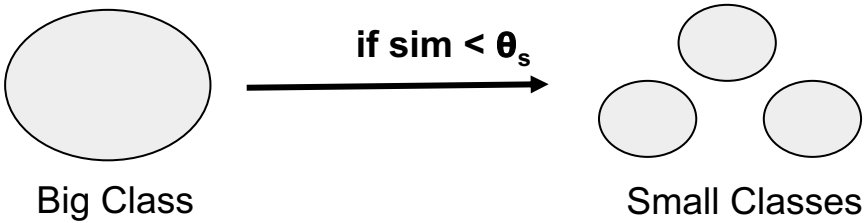
- Spliting and merging classes with attribute similarity

$$Sim(A, B) = \frac{2(|A \cap B|)}{|A|+|B|}$$

Object Clustering

- Spliting and merging classes with attribute similarity

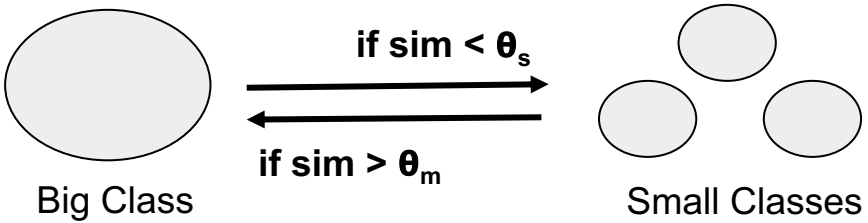
$$Sim(A, B) = \frac{2(|A \cap B|)}{|A|+|B|}$$



Object Clustering

- Spliting and merging classes with attribute similarity

$$Sim(A, B) = \frac{2(|A \cap B|)}{|A|+|B|}$$



Relationship Inference

- Run & Record

Relationship Inference

- Run & Record

```
1  0  obj  …  endobj
2  0  obj  …  endobj
3  0  obj  …  endobj

4  0  obj  …  endobj
5  0  obj  …  endobj
6  0  obj  …  endobj
```

Relationship Inference

- Run & Record

```
1  0  obj  …  endobj
2  0  obj  …  endobj
3  0  obj  …  endobj

4  0  obj  …  endobj
5  0  obj  …  endobj
6  0  obj  …  endobj
```

```
try{
   var annots = this.getAnnot();
   app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

Relationship Inference

- Run & Record

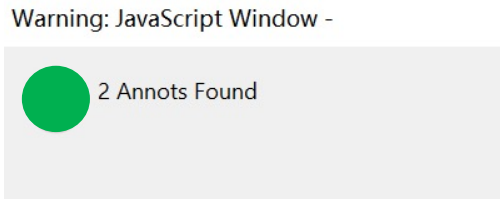
```
1  0  obj  …  endobj
2  0  obj  …  endobj
3  0  obj  …  endobj
```

```
try{
   var annots = this.getAnnot();
   app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4  0  obj  …  endobj
5  0  obj  …  endobj
6  0  obj  …  endobj
```

Relationship Inference

- Run & Record



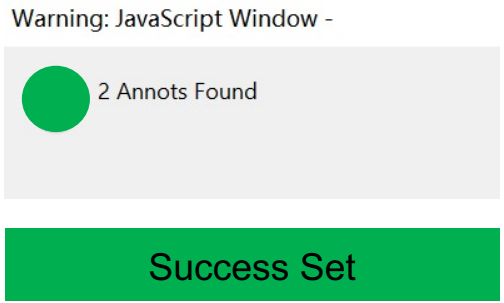

```
1  0  obj  …  endobj
2  0  obj  …  endobj
3  0  obj  …  endobj
```

```
try{
   var annots = this.getAnnot();
   app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4  0  obj  …  endobj
5  0  obj  …  endobj
6  0  obj  …  endobj
```

Relationship Inference

- Run & Record



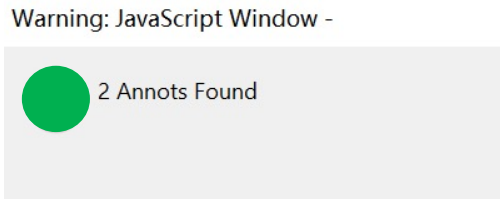

```
1  0  obj  …  endobj
2  0  obj  …  endobj
3  0  obj  …  endobj
```

```
try{
   var annots = this.getAnnot();
   app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

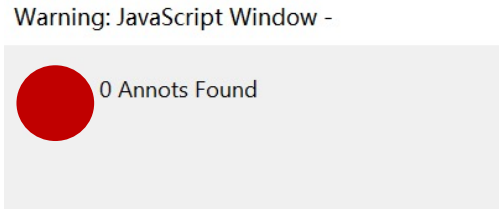
```
4  0  obj  …  endobj
5  0  obj  …  endobj
6  0  obj  …  endobj
```

Relationship Inference

- Run & Record



Success Set



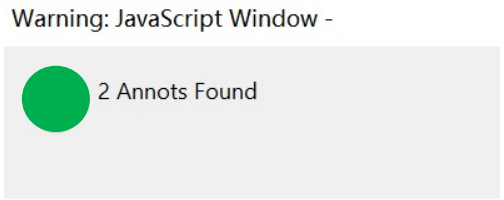

```
1  0  obj  …  endobj
2  0  obj  …  endobj
3  0  obj  …  endobj
```

```
try{
   var annots = this.getAnnot();
   app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

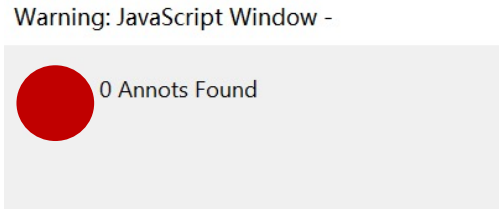
```
4  0  obj  …  endobj
5  0  obj  …  endobj
6  0  obj  …  endobj
```

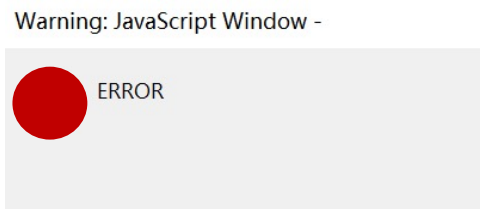
Relationship Inference

- Run & Record



Success Set






```
1  0  obj  …  endobj
2  0  obj  …  endobj
3  0  obj  …  endobj
```

```
try{
   var annots = this.getAnnot();
   app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4  0  obj  …  endobj
5  0  obj  …  endobj
6  0  obj  …  endobj
```

Relationship Inference

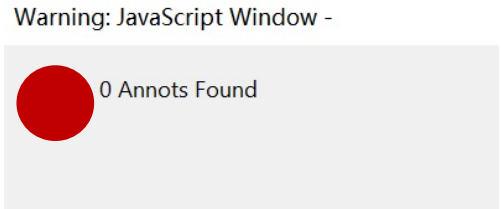
- Run & Record

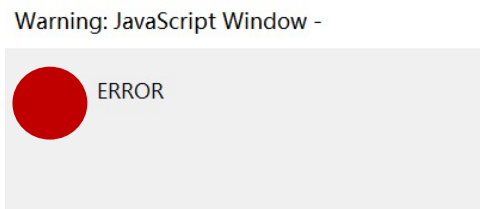
Success Set

```
1 0 obj … endobj
2 0 obj … endobj
3 0 obj … endobj
```

```
try{
   var annots = this.getAnnot();
   app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4 0 obj … endobj
5 0 obj … endobj
6 0 obj … endobj
```





Failure Set

Relationship Inference

- Run & Record

Warning: JavaScript Window -

2 Annots Found

Success Set

```
1  0  obj  …  endobj
2  0  obj  …  endobj
3  0  obj  …  endobj
```

```
try{
   var annots = this.getAnnot();
   app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4  0  obj  …  endobj
5  0  obj  …  endobj
6  0  obj  …  endobj
```

Warning: JavaScript Window -

0 Annots Found

Warning: JavaScript Window -

ERROR

Failure Set

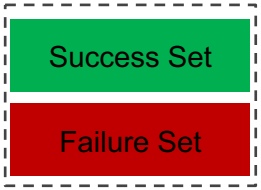
Samples

Success Set

Failure Set

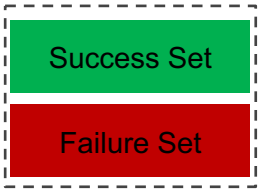
Relationship Inference

- Run & Record
- Statistical Inference



Relationship Inference

- Run & Record
- Statistical Inference



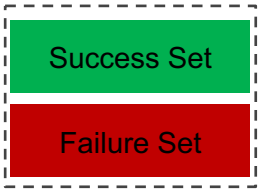
Statistical Inference

	API_0			API_1			API_2			…
	Success Rate	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**…**
Class0	0.9	0.1	0.8	1.0	0.0	1.0	1.0	0.1	0.9	…
Class1	0.8	0.2	0.6	0.3	0.3	0.0	0.9	0.0	0.9	…
Class2	1.0	0.0	1.0	0.9	0.1	0.8	0.2	0.1	0.1	…
…	…	…	…	…	…	…	…	…	…	…
Classn	0.3	0.3	0.0	0.8	0.2	0.6	1.0	0.2	0.8	…

Relation Map

Relationship Inference

- Run & Record
- Statistical Inference



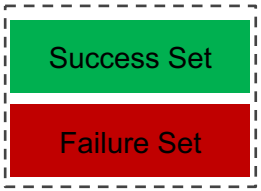
Statistical Inference

	API_0			API_1			API_2			...
	Success Rate	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**...**
Class0	0.9	0.1	0.8	1.0	0.0	1.0	1.0	0.1	0.9	...
Class1	0.8	0.2	0.6	0.3	0.3	0.0	0.9	0.0	0.9	...
Class2	1.0	0.0	1.0	0.9	0.1	0.8	0.2	0.1	0.1	...
...	...	...	...	...	...	...	...	...	...	...
Classn	0.3	0.3	0.0	0.8	0.2	0.6	1.0	0.2	0.8	...

Relation Map

Relationship Inference

- Run & Record
- Statistical Inference



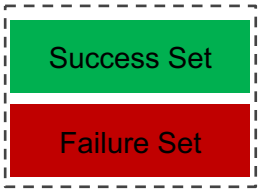
Statistical Inference

	API_0			API_1			API_2			…
	Success Rate	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**…**
Class0	0.9	0.1	0.8	1.0	0.0	1.0	1.0	0.1	0.9	…
Class1	0.8	0.2	0.6	0.3	0.3	0.0	0.9	0.0	0.9	…
Class2	1.0	0.0	1.0	0.9	0.1	0.8	0.2	0.1	0.1	…
…	…	…	…	…	…	…	…	…	…	…
Classn	0.3	0.3	0.0	0.8	0.2	0.6	1.0	0.2	0.8	…

Relation Map

Relationship Inference

- Run & Record
- Statistical Inference



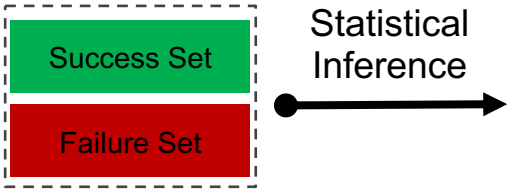
Statistical Inference

	API_0			API_1			API_2			…
	Success Rate	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**…**
Class0	0.9	0.1	0.8	1.0	0.0	1.0	1.0	0.1	0.9	…
Class1	0.8	0.2	0.6	0.3	0.3	0.0	0.9	0.0	0.9	…
Class2	1.0	0.0	1.0	0.9	0.1	0.8	0.2	0.1	0.1	…
…	…	…	…	…	…	…	…	…	…	…
Classn	0.3	0.3	0.0	0.8	0.2	0.6	1.0	0.2	0.8	…

Relation Map

Relationship Inference

- Run & Record
- Statistical Inference



	API_0			API_1			API_2			...
	Success Rate	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**Success Rate**	**Failure Rate**	**Diff**	**...**
Class0	0.9	0.1	0.8	1.0	0.0	1.0	1.0	0.1	0.9	...
Class1	0.8	0.2	0.6	0.3	0.3	0.0	0.9	0.0	0.9	...
Class2	1.0	0.0	1.0	0.9	0.1	0.8	0.2	0.1	0.1	...
...	...	...	...	...	...	...	...	...	...	...
Classn	0.3	0.3	0.0	0.8	0.2	0.6	1.0	0.2	0.8	...

Relation Map

Relationship Guided Mutation

	API_0	**API_1**	**API_2**	**…**
Class0	0.8	1.0	0.9	…
Class1	0.6	0.0	0.9	…
Class2	1.0	0.8	0.1	…
…	**…**	**…**	**…**	**…**

Relation Map

Relationship Guided Mutation

	API_0	**API_1**	**API_2**	**…**
Class0	0.8	1.0	0.9	…
Class1	0.6	0.0	0.9	…
Class2	1.0	0.8	0.1	…
…	**…**	**…**	**…**	**…**

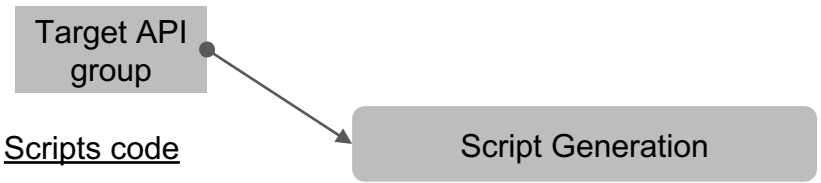
Relation Map

Target API group

Relationship Guided Mutation

	API_0	API_1	API_2	…
Class0	0.8	1.0	0.9	…
Class1	0.6	0.0	0.9	…
Class2	1.0	0.8	0.1	…
…	**…**	**…**	**…**	**…**

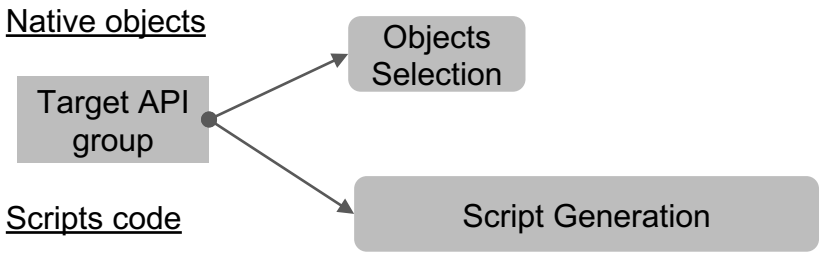
Relation Map



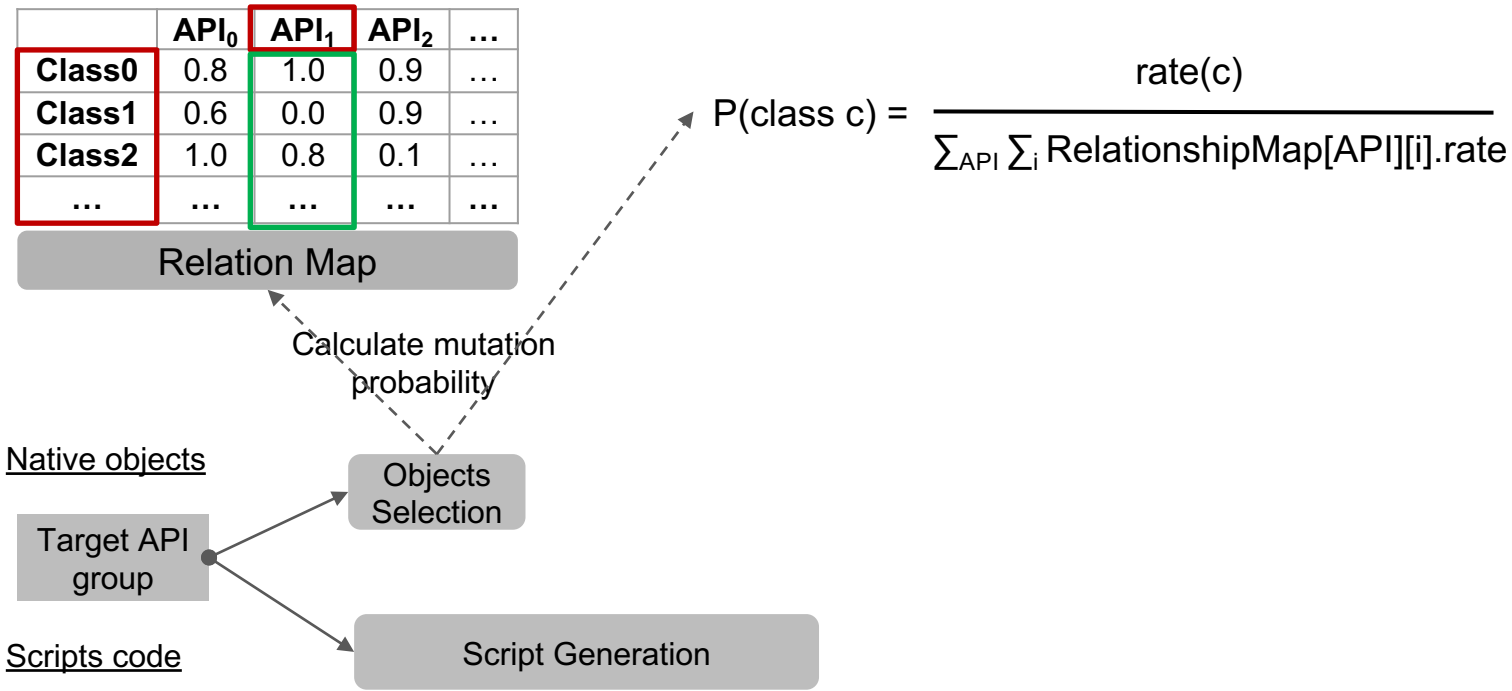
Relationship Guided Mutation

	API_0	API_1	API_2	…
Class0	0.8	1.0	0.9	…
Class1	0.6	0.0	0.9	…
Class2	1.0	0.8	0.1	…
…	**…**	**…**	**…**	**…**

Relation Map



Relationship Guided Mutation



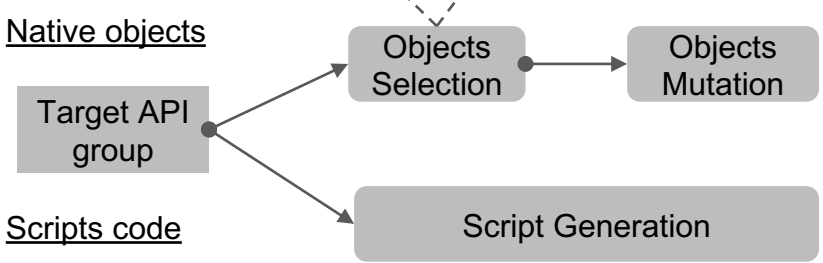
Relationship Guided Mutation

	API_0	API_1	API_2	…
Class0	0.8	1.0	0.9	…
Class1	0.6	0.0	0.9	…
Class2	1.0	0.8	0.1	…
…	…	…	…	…

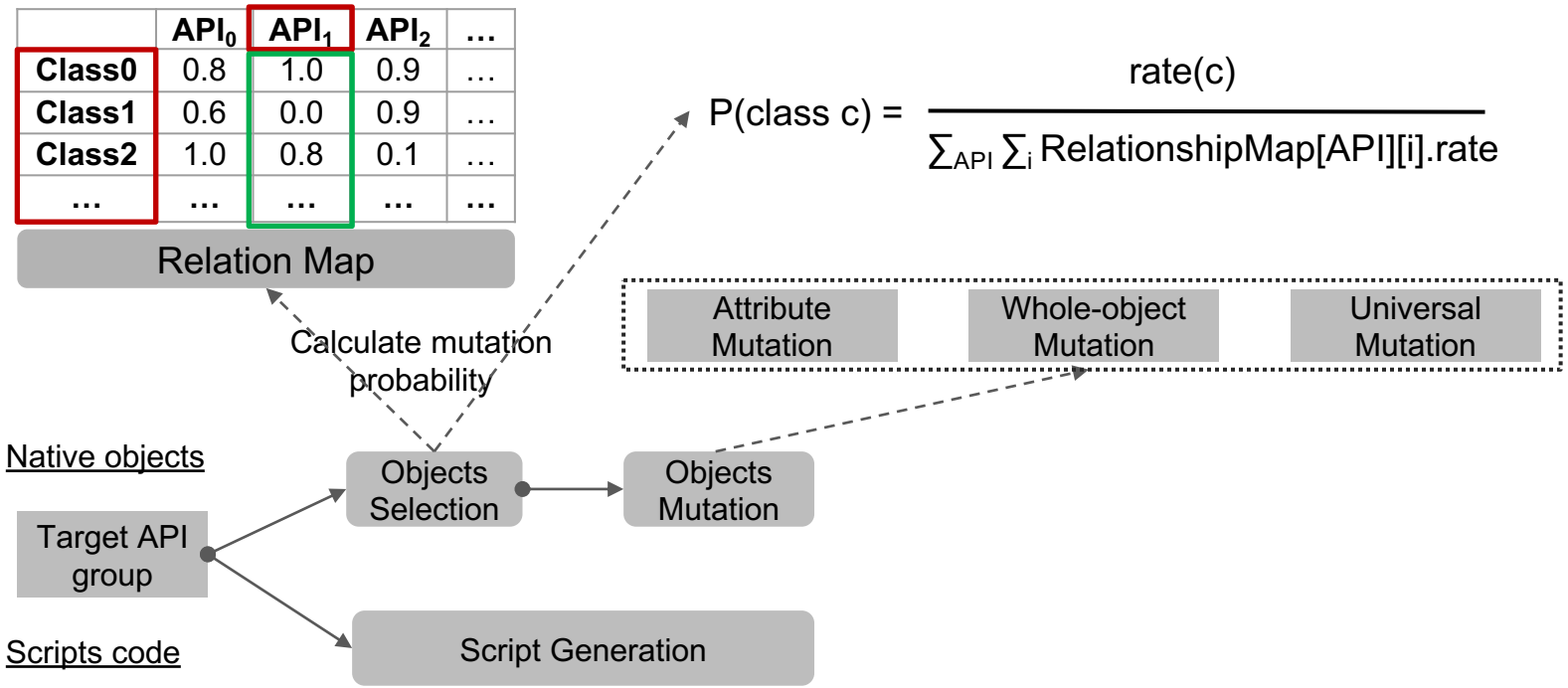
Relation Map

$$P(\text{class } c) = \frac{\text{rate}(c)}{\sum_{API} \sum_{i} \text{RelationshipMap}[API][i].\text{rate}}$$

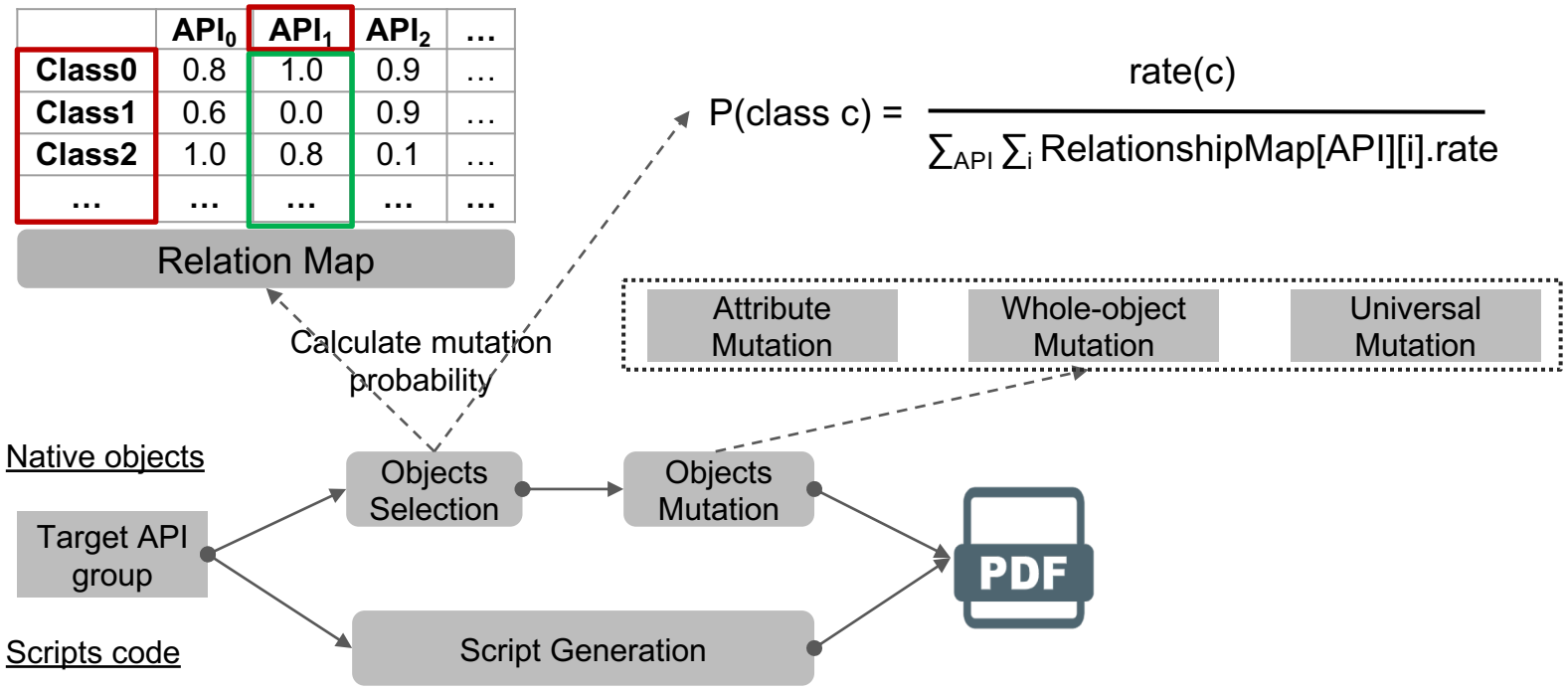
Calculate mutation probability



Relationship Guided Mutation



Relationship Guided Mutation



Implementation

- 4.3K lines of code in Python

Implementation

- 4.3K lines of code in Python
- Currently support
  - PDF: use PyPDF2 for parsing
  - Word: use zipfile and xml for parsing

Implementation

- 4.3K lines of code in Python
- Currently support
  - PDF: use PyPDF2 for parsing
  - Word: use zipfile and xml for parsing
- For scripts generation
  - modify Domato, and add block-level template

Implementation

- 4.3K lines of code in Python
- Currently support
  - PDF: use PyPDF2 for parsing
  - Word: use zipfile and xml for parsing
- For scripts generation
  - modify Domato, and add block-level template
- Extensible and Portable

Evaluations

- New bugs
- Comparison
  - Bug finding
  - Branch coverage

New Bugs (in four months)

	Adobe Acrobat	Foxit Reader	Microsoft Word	**Total**
use-after-free	12	18	3	**33**
buffer overflow	4	8	5	**17**
buffer error	6	1	0	**7**
null ptr deref	30	22	8	**60**
stack exhaustion	6	4	0	**10**
access violation	2	2	1	**5**
others	0	1	1	**2**
Total	**60**	**56**	**18**	**134**

New Bugs (in four months)

	Adobe Acrobat	Foxit Reader	Microsoft Word	**Total**
use-after-free	12	18	3	**33**
buffer overflow	4	8	5	**17**
buffer error	6	1	0	**7**
null ptr deref	30	22	8	**60**
stack exhaustion	6	4	0	**10**
access violation	2	2	1	**5**
others	0	1	1	**2**
Total	**60**	**56**	**18**	**134**

New Bugs (in four months)

	Adobe Acrobat	Foxit Reader	Microsoft Word	**Total**
use-after-free	12	18	3	**33**
buffer overflow	4	8	5	**17**
buffer error	6	1	0	**7**
null ptr deref	30	22	8	**60**
stack exhaustion	6	4	0	**10**
access violation	2	2	1	**5**
others	0	1	1	**2**
Total	**60**	**56**	**18**	**134**

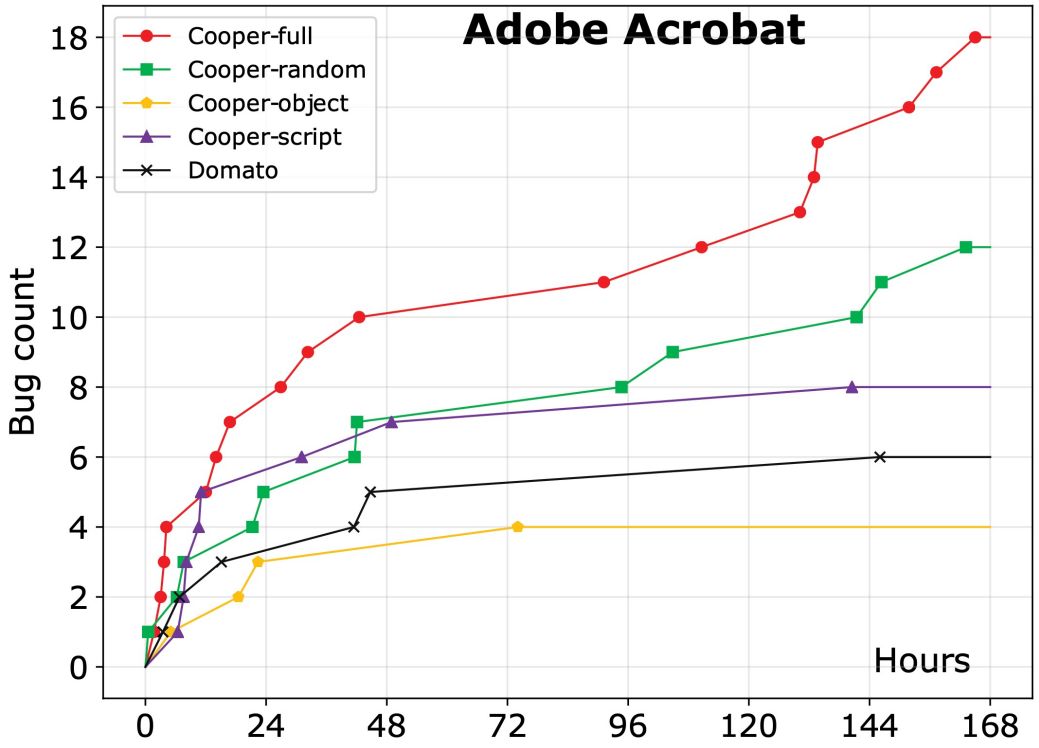
- 33 CVE, 59 fixed
- 22K dollars bounty
- 90 APIs &

  11 object classes

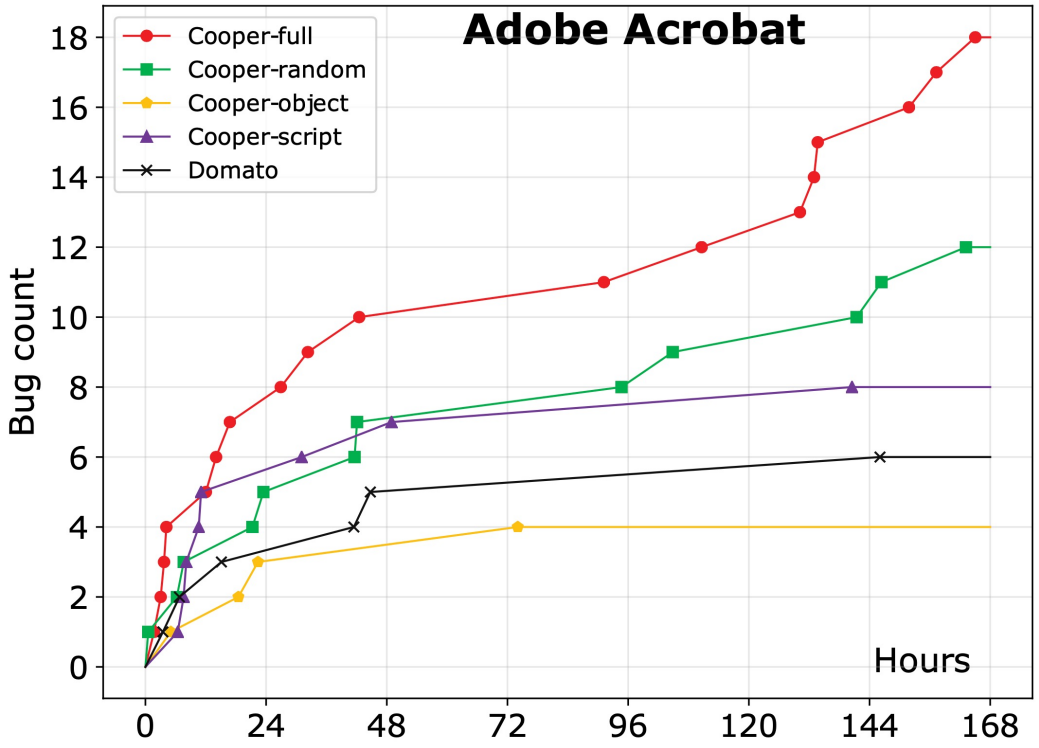
Comparing different configurations and tools

Experiment	Relation Guidance	Object Mutation	Script Generation
Cooper-full	●	●	●
Cooper-random	○	●	●
Cooper-object	●	●	○
Cooper-script	○	○	●
Domato	○	○	◐

Bug finding with different configurations (one week)

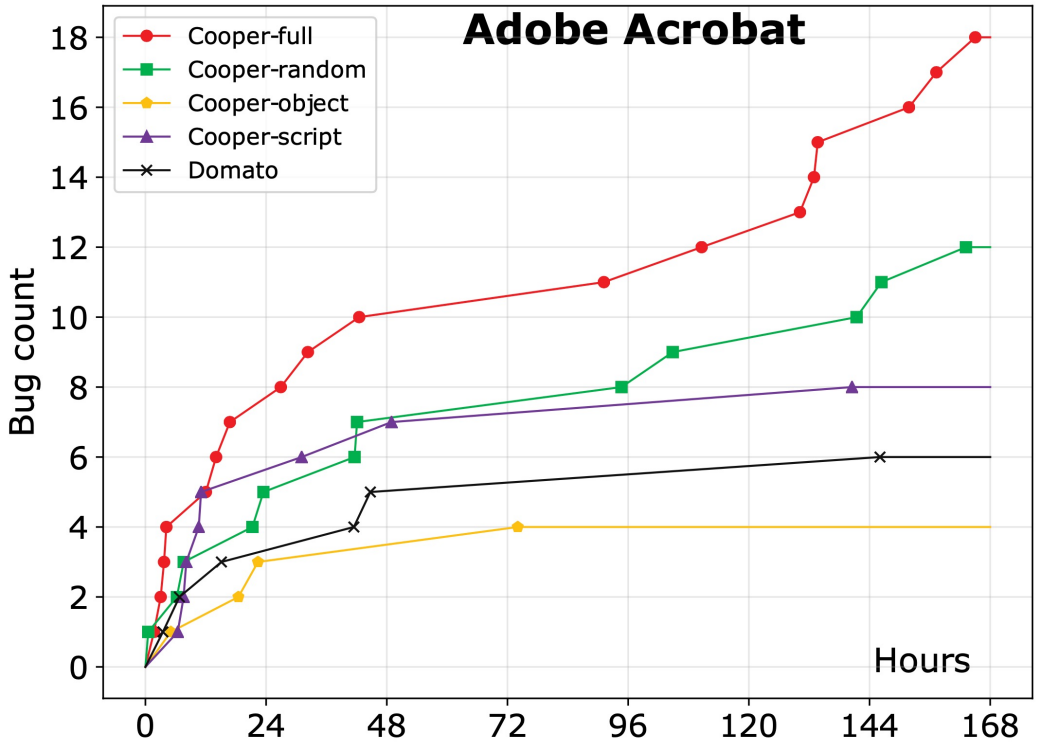


Bug finding with different configurations (one week)



Cooper-full: 18 bugs

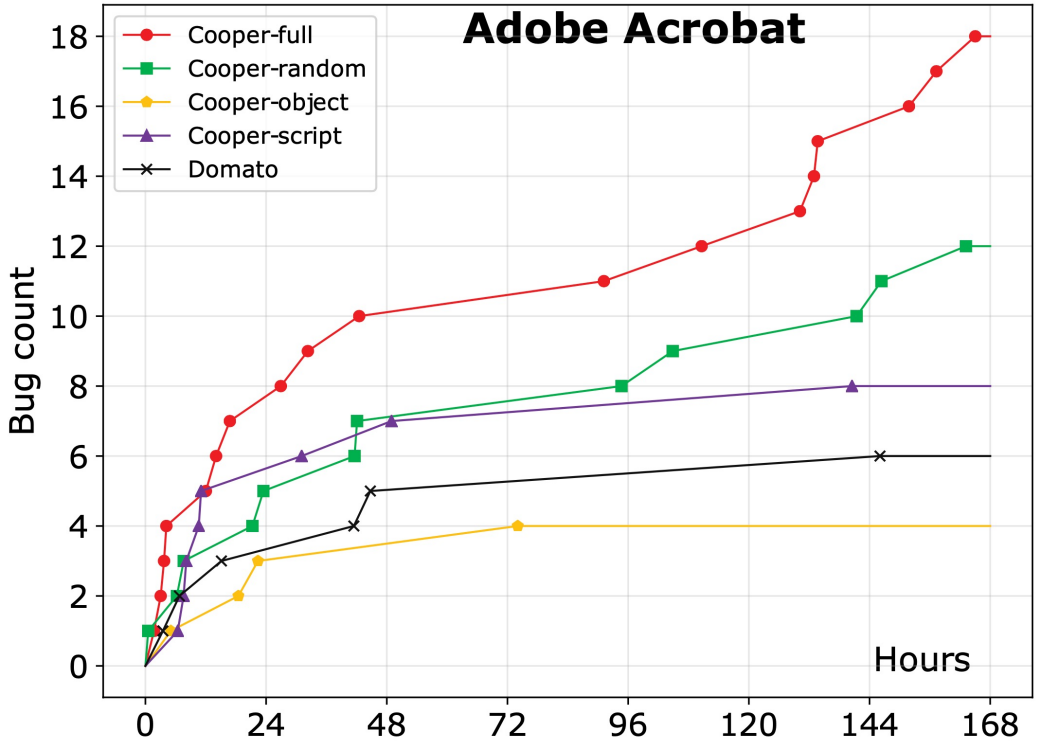
Bug finding with different configurations (one week)



Cooper-full: 18 bugs

w/o relation: 12 bugs

Bug finding with different configurations (one week)

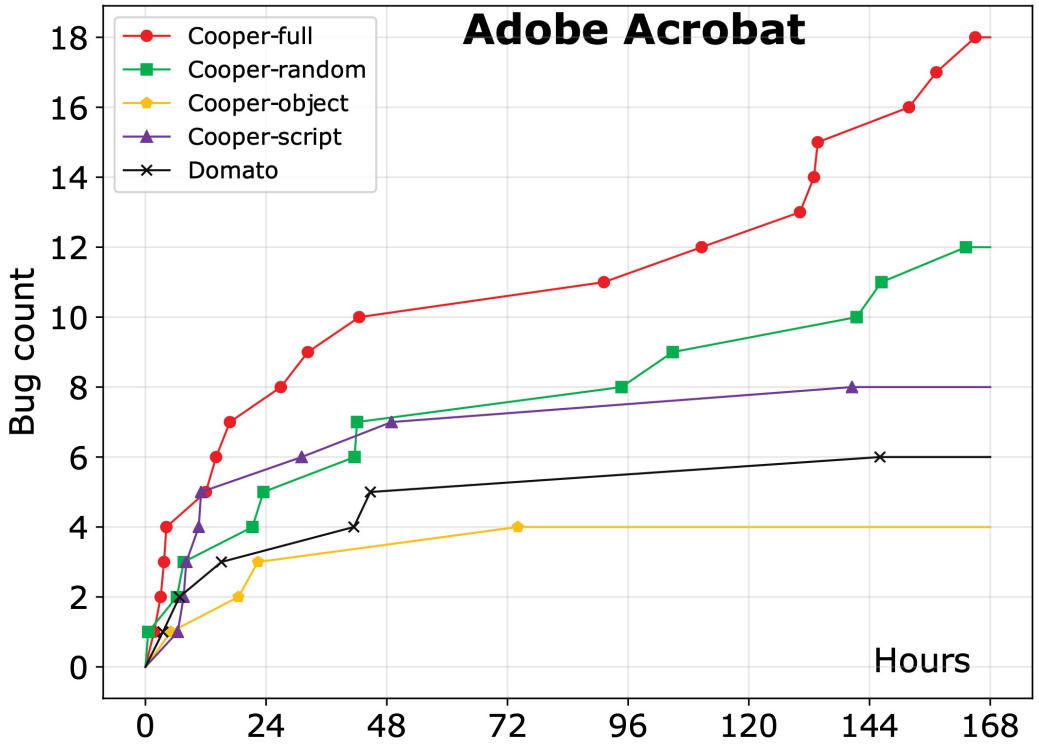


Cooper-full: 18 bugs

w/o relation: 12 bugs

Script-only: 8 bugs

Bug finding with different configurations (one week)



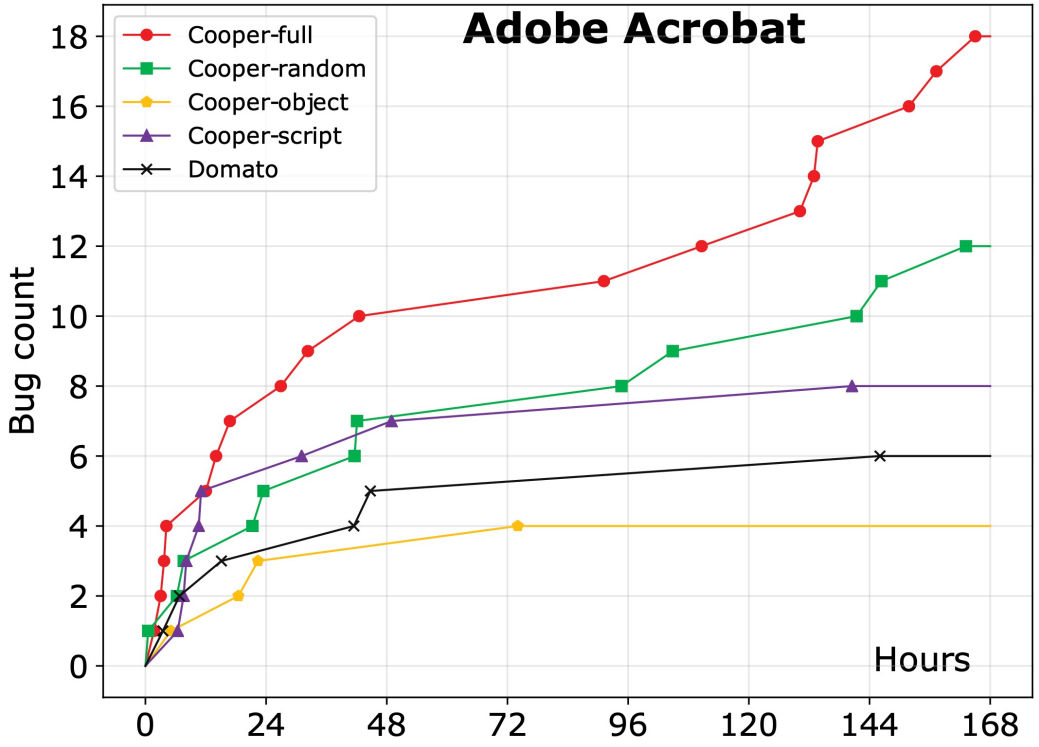
Cooper-full: 18 bugs

w/o relation: 12 bugs

Script-only: 8 bugs

Domato: 6 bugs

Bug finding with different configurations (one week)



Cooper-full: 18 bugs

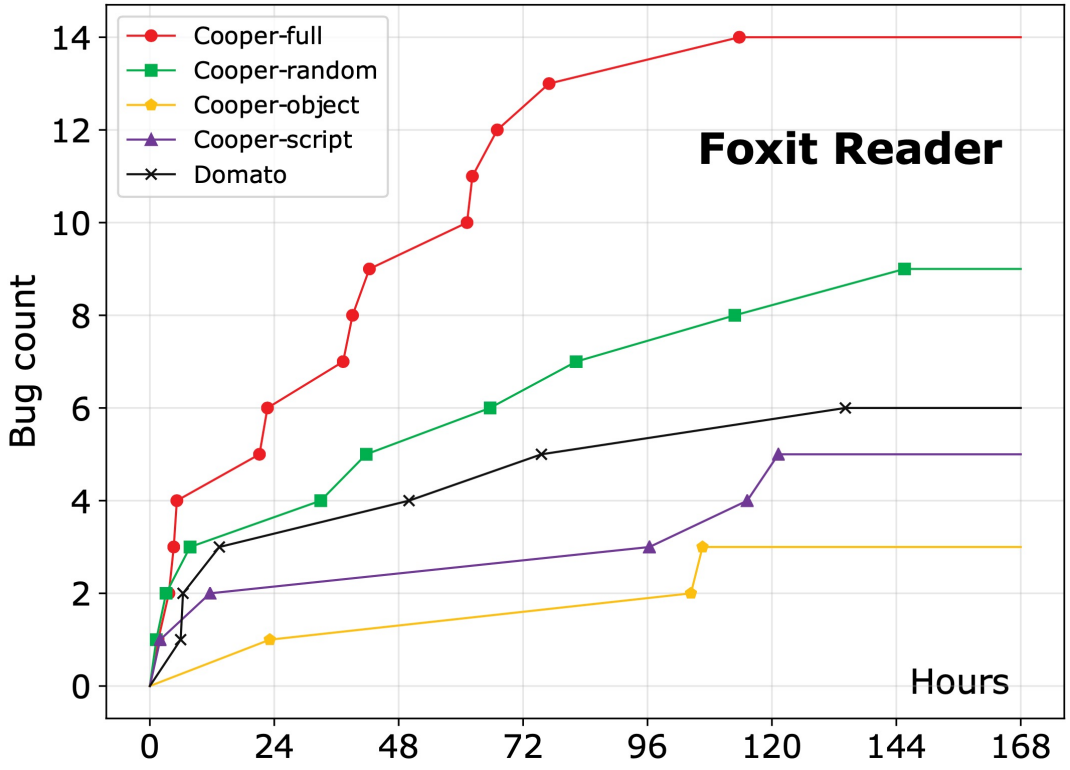
w/o relation: 12 bugs

Script-only: 8 bugs

Domato: 6 bugs

Object-only: 4 bugs

Bug finding with different configurations (one week)



Cooper-full: 14 bugs

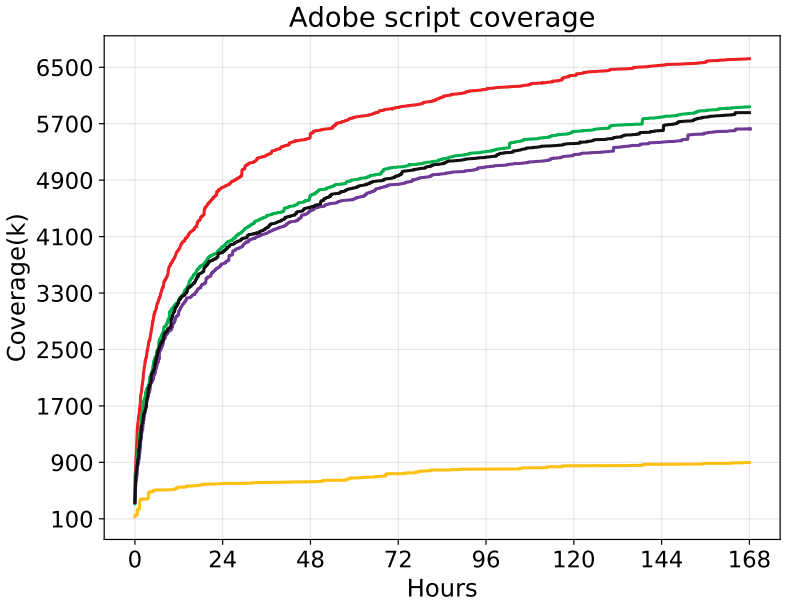
w/o relation: 9 bugs

Domato: 6 bugs

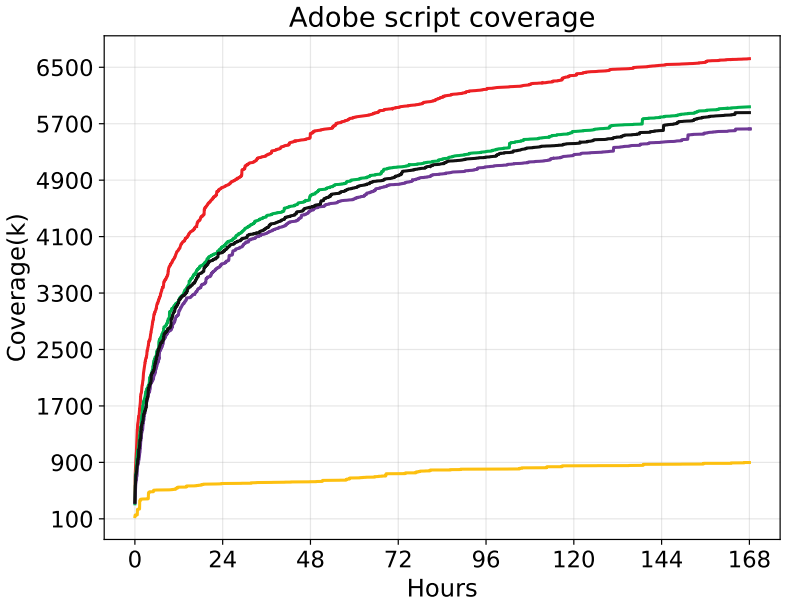
Script-only: 5 bugs

Object-only: 3 bugs

Branch Coverage (one week)

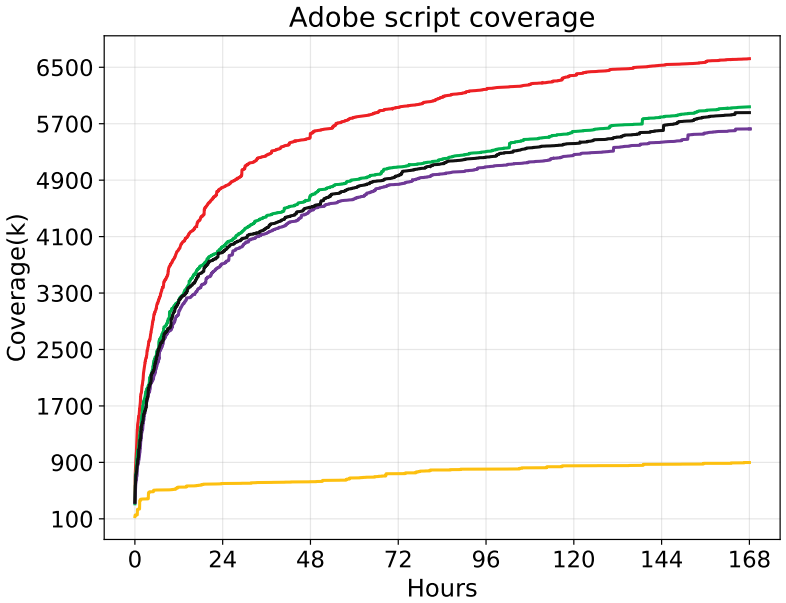


Branch Coverage (one week)



Cooper-full

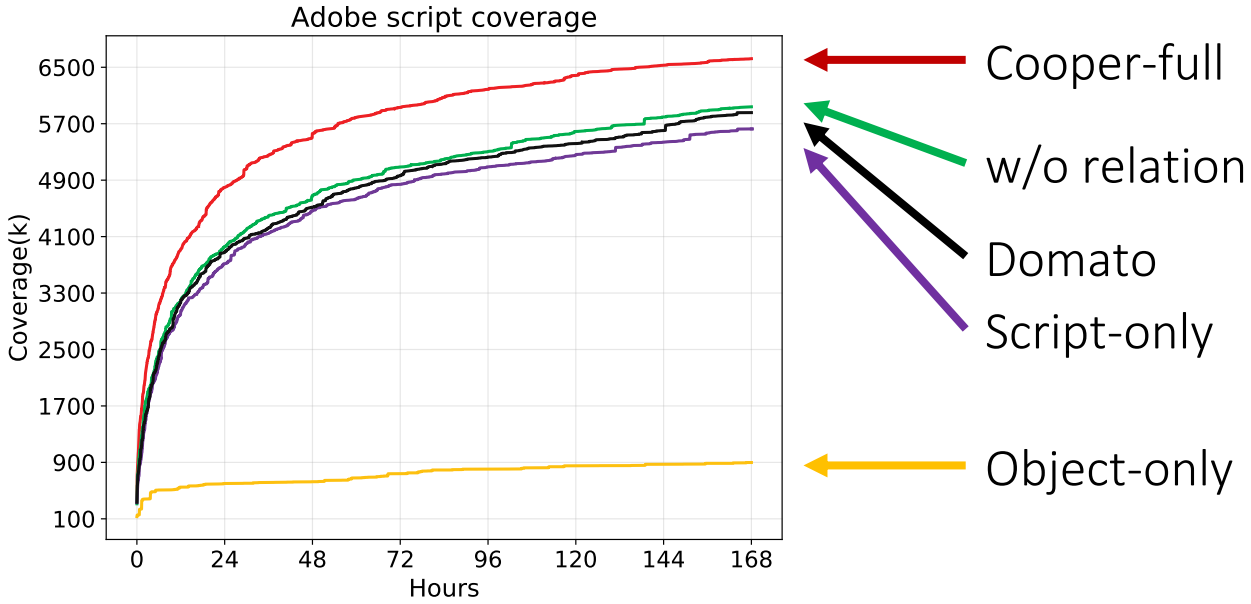
Branch Coverage (one week)



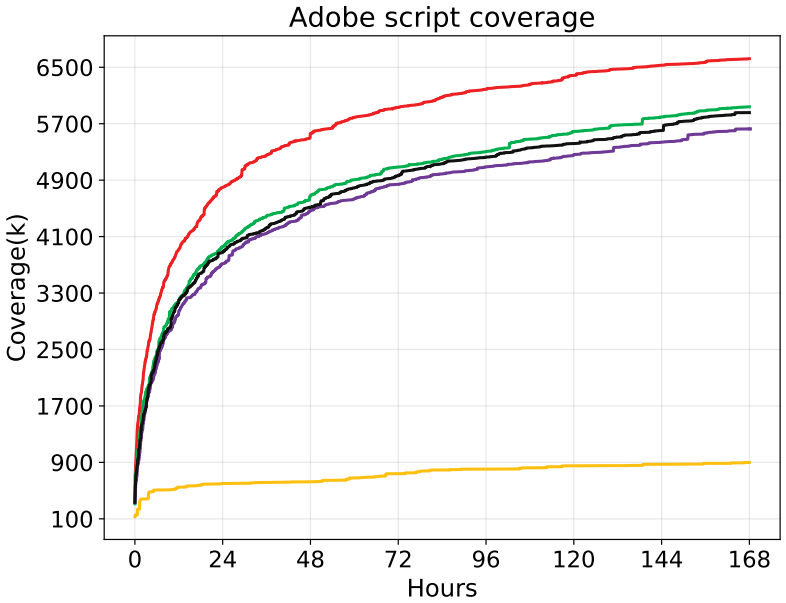
Cooper-full

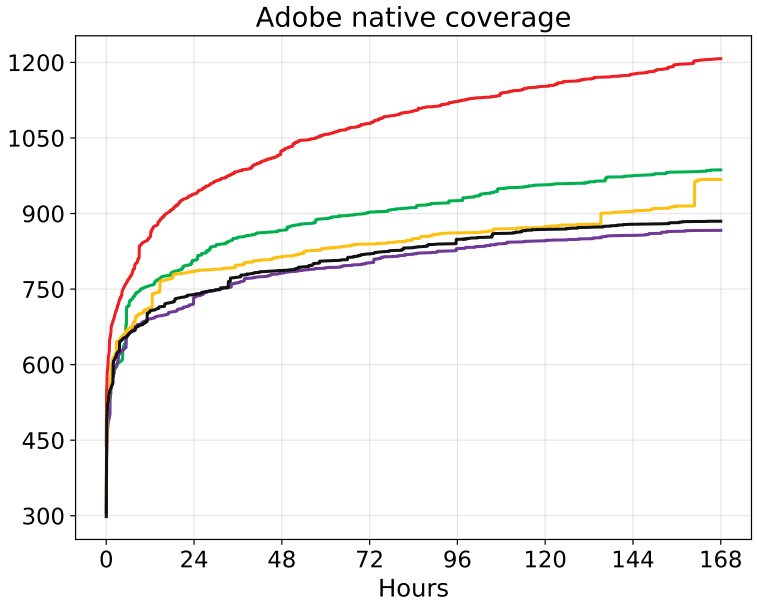
Object-only

Branch Coverage (one week)

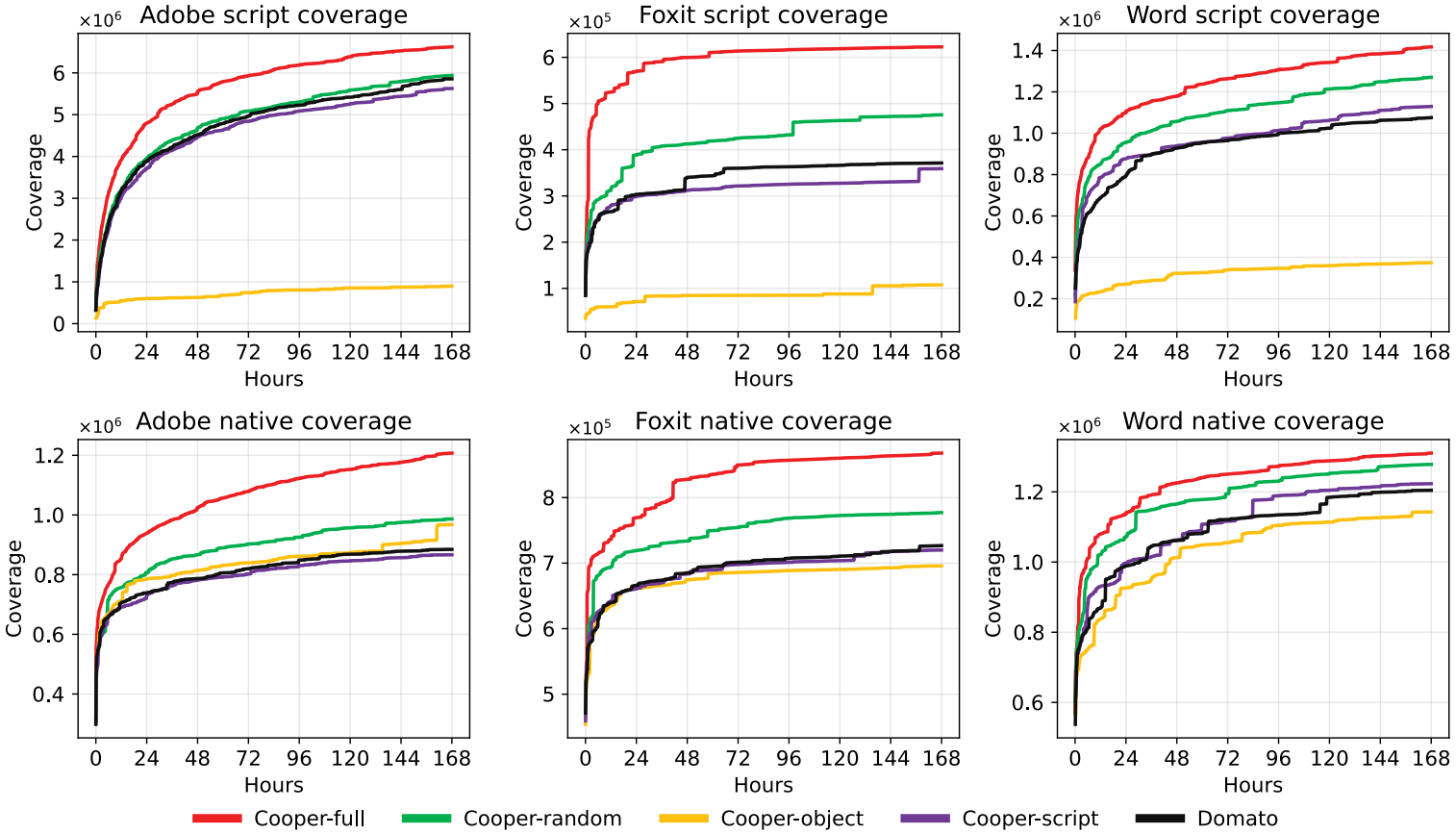


Branch Coverage (one week)





Branch Coverage (one week)



Conclusion

Conclusion

- Cooperative mutation
  - effectively test binding code of scripting languages

Conclusion

- Cooperative mutation
  - effectively test binding code of scripting languages
- 134 bugs in Adobe Acrobat, Foxit Reader, and Microsoft Word
  - 33 CVE and 22K dollars bounty

Conclusion

- Cooperative mutation
  - effectively test binding code of scripting languages
- 134 bugs in Adobe Acrobat, Foxit Reader, and Microsoft Word
  - 33 CVE and 22K dollars bounty
- Code at: https://github.com/TCA-ISCAS/Cooper

Question?