a place of mind
THE UNIVERSITY OF BRITISH COLUMBIA

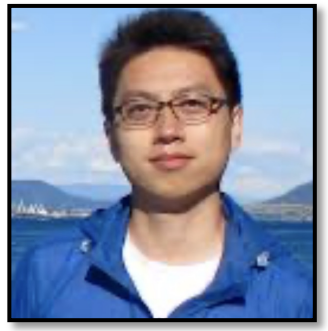
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Responsibility in Context: On Applicability of Slicing in Semantic Regression Analysis

ICSE 2023

Sahar Badihi
UBC, Canada

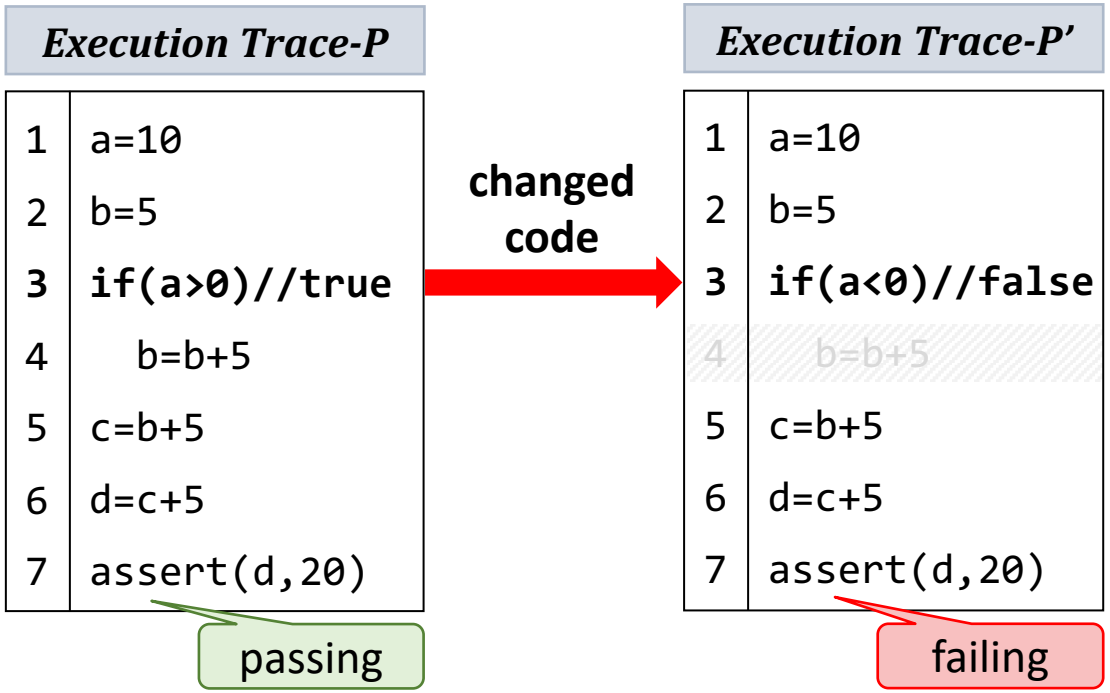
Khaled Ahmed
UBC, Canada



Yi Li
NTU, Singapore

Julia Rubin
UBC, Canada

Regressions Are Common



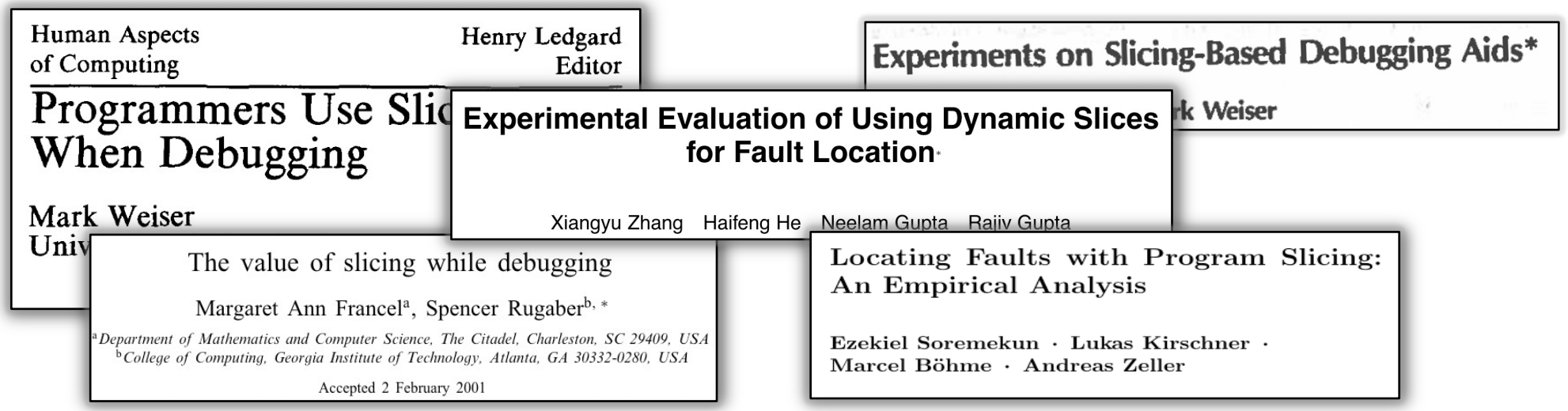
Regressions: **unintended** behavioral differences between the original and modified versions of the program

76% of code upgrades result in regressions [*]

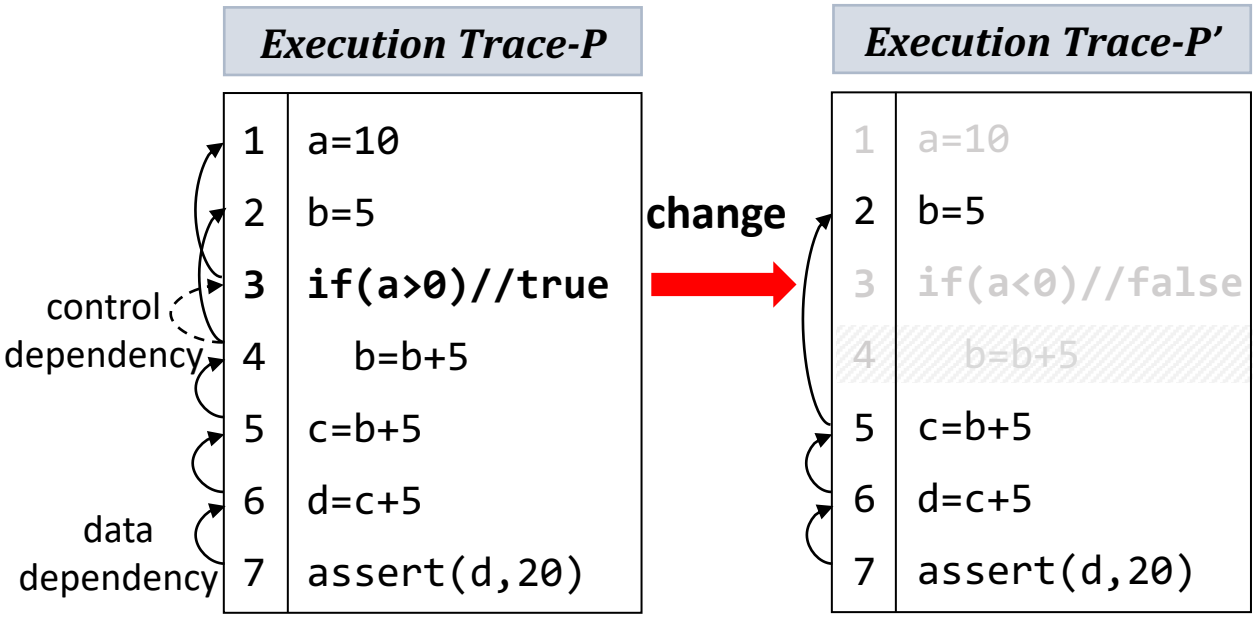
[*] M. Shaikh, R. Rodriguez, and X. Wang. "*Experience paper: a study on behavioral backward incompatibilities of Java software libraries*", ISSTA'17.

Slicing-based Approaches for Troubleshooting Regressions

- Identify *sequences* of statements affecting a failure
- Aligned with the mental-model developers use



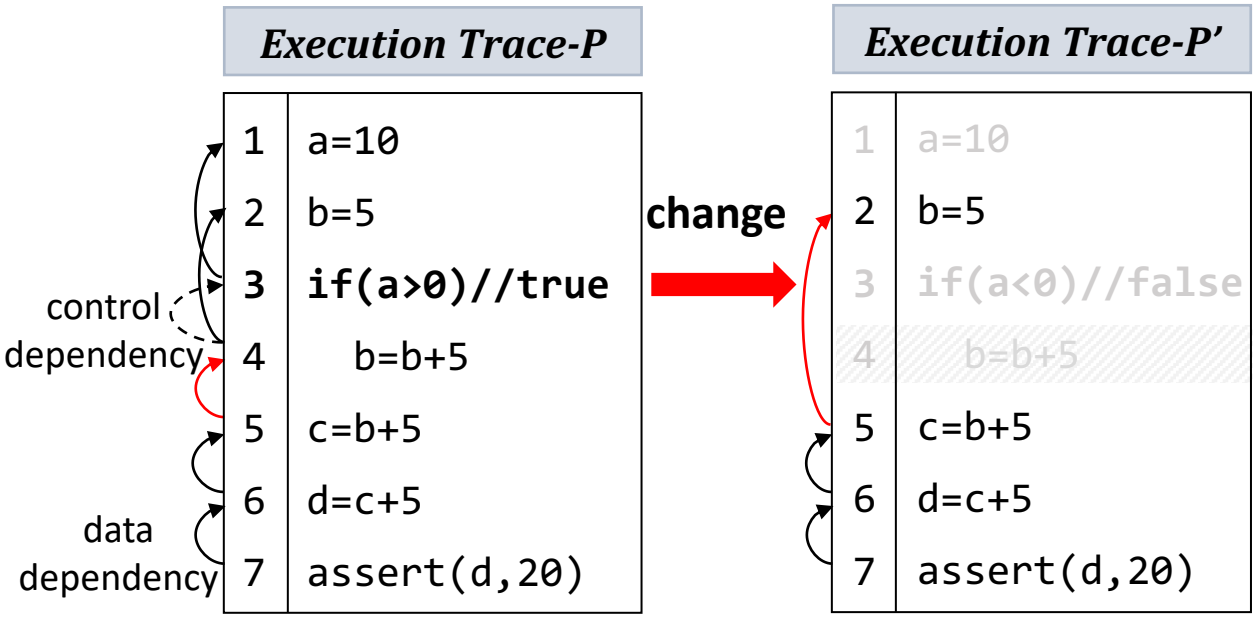
Classical Slicing



→	s2 is **data-dependent** on s1:	s1 defines a variable used in s2
-->	s2 is **control-dependent** on s1:	s1 directly affects whether s2 is executed

[*] D. Weeratunge, X. Zhang, W. N. Sumner, and S. Jagannathan, "*Analyzing Concurrency Bugs Using Dual Slicing*", ISSTA'10.

Classical Slicing



→	s2 is **data-dependent** on s1:	s1 defines a variable used in s2
->	s2 is **control-dependent** on s1:	s1 directly affects whether s2 is executed

[*] D. Weeratunge, X. Zhang, W. N. Sumner, and S. Jagannathan, "*Analyzing Concurrency Bugs Using Dual Slicing*", ISSTA'10.

Dual (Symmetric) Slicing

Execution Trace-P			Execution Trace-P'	
1	a=10	identical	1	a=10
2	b=5	identical	2	b=5
3	**if(a>0)//true**		**3**	**if(a<0)//false**
4	b=b+5		4	b=b+5
5	c=b+5	data diff.	5	c=b+5
6	d=c+5	data diff.	6	d=c+5
7	assert(d,20)	data diff.	7	assert(d,20)

Identifies **aligned** trace execution statement (same code statement under the same control path)

[*] D. Weeratunge, X. Zhang, W. N. Sumner, and S. Jagannathan, "*Analyzing Concurrency Bugs Using Dual Slicing*", ISSTA'10.

Dual (Symmetric) Slicing

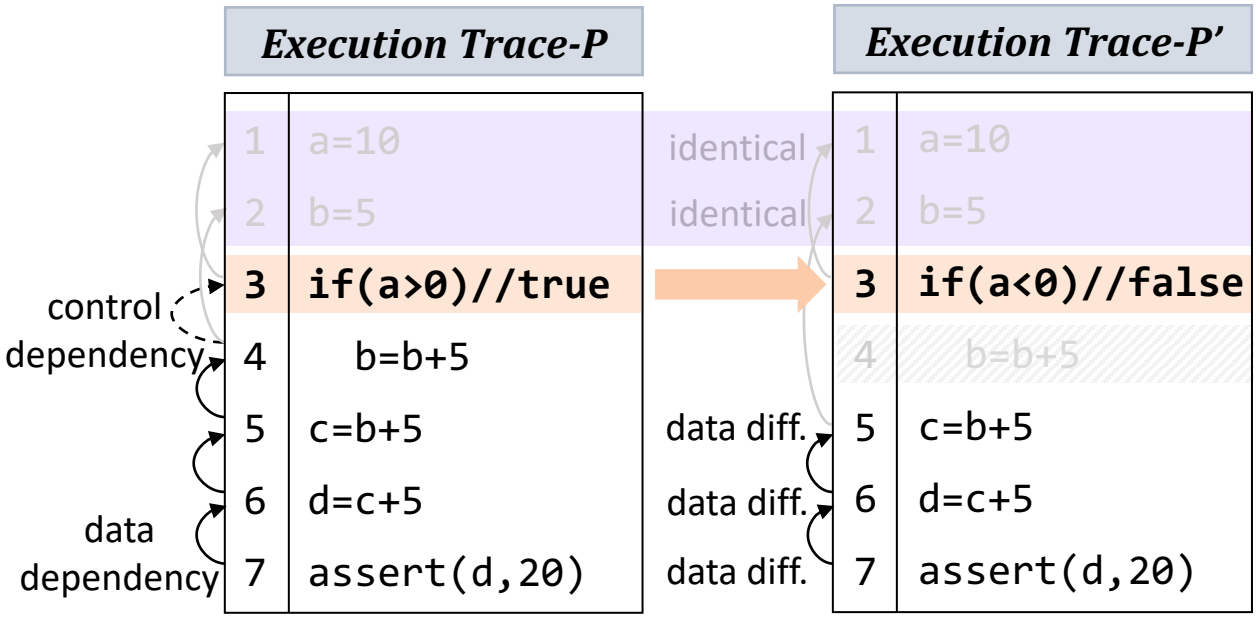
Execution Trace-P			Execution Trace-P'	
1	a=10	identical	1	a=10
2	b=5	identical	2	b=5
3	if(a>0)//true		3	if(a<0)//false
4	b=b+5		4	b=b+5
5	c=b+5	data diff.	5	c=b+5
6	d=c+5	data diff.	6	d=c+5
7	assert(d,20)	data diff.	7	assert(d,20)

Idea 1: focus only on differences, ignoring aligned statements with the same control and data values

Align trace execution statement (same statement under the same control path)

[*] D. Weeratunge, X. Zhang, W. N. Sumner, and S. Jagannathan, "*Analyzing Concurrency Bugs Using Dual Slicing*", ISSTA'10.

Dual (Symmetric) Slicing



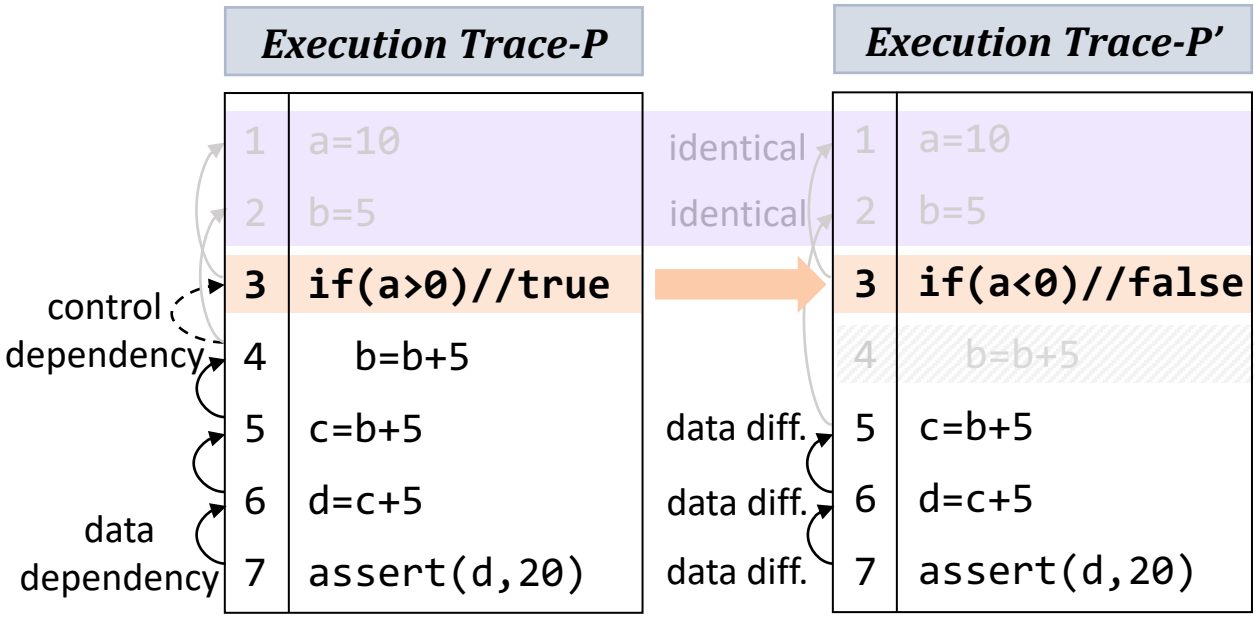
Idea 1: focus only on differences, ignoring aligned statements with the same control and data values

Idea 2: synchronize slices by adding aligned trace statement (and their transitive dependencies) to each slice

→	s2 is **data-dependent** on s1:	s1 defines a variable used in s2
->	s2 is **control-dependent** on s1:	s1 directly affects whether s2 is executed

[*] D. Weeratunge, X. Zhang, W. N. Sumner, and S. Jagannathan, "*Analyzing Concurrency Bugs Using Dual Slicing*", ISSTA'10.

Dual (Symmetric) Slicing



Idea 1: focus only on differences, ignoring aligned statements with the same control and data values

Idea 2: synchronize slices by adding aligned trace statement (and their transitive dependencies) to each slice

Produces more precise slices (e.g., handles omission errors)

[*] D. Weeratunge, X. Zhang, W. N. Sumner, and S. Jagannathan, "*Analyzing Concurrency Bugs Using Dual Slicing*", ISSTA'10.

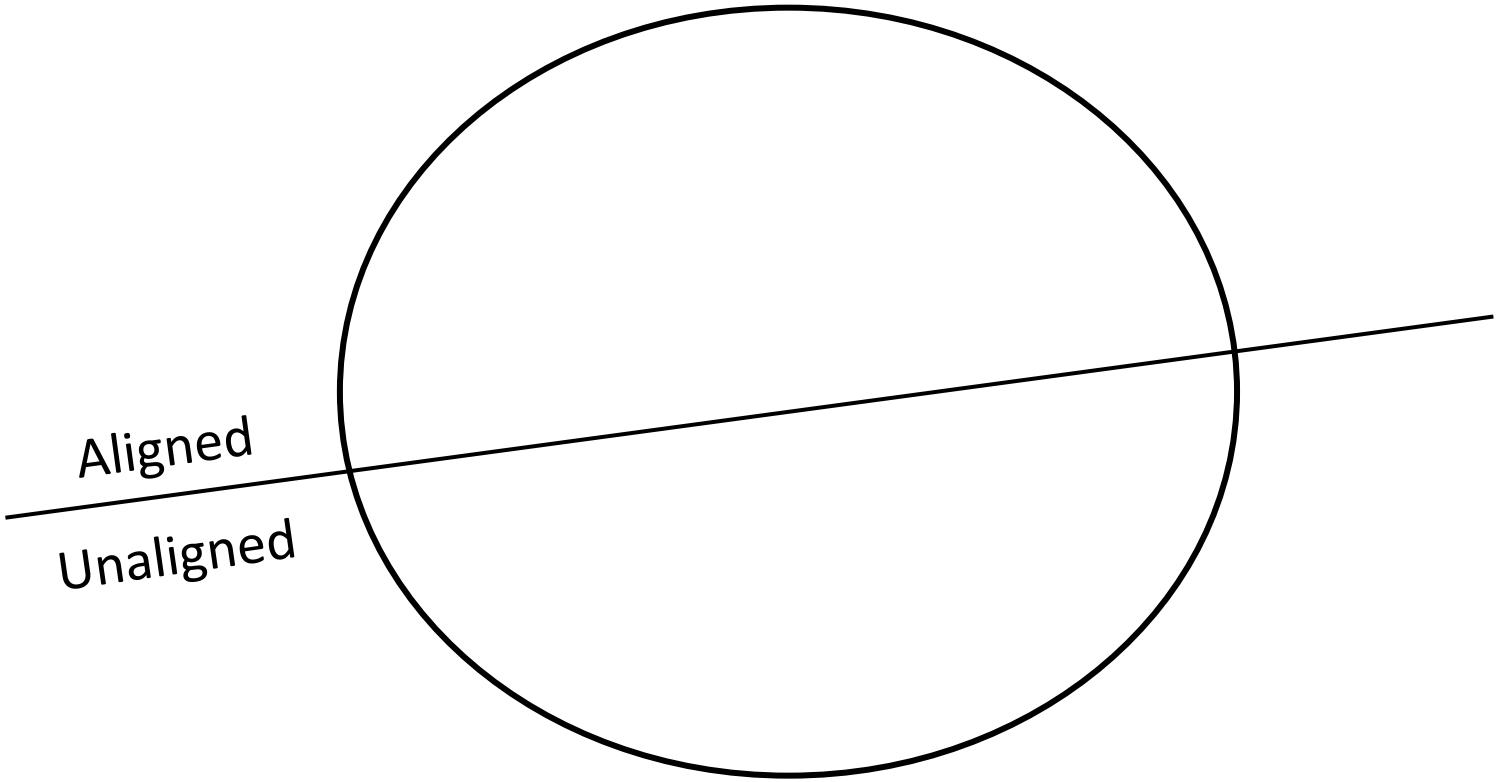
Study: Slices are Too Long and Unfocused

	Project	Failures	LoC	Trace	DSlice
Defects4J	Chart	23	96,517	5,912	61
	Closure	95	90,601	92,663	3,341
	Lang	49	22,750	3,091	43
	Math	63	85,617	8,801	738
	Mockito	26	37,277	3,114	377
	Time	22	28,422	13,037	518
	Average	**-**	**60,199**	**36,017**	**1,398**
LibRench: Client-Library	jettison / xstream	1	63,631	21,502	515
	square-dagger / modelmapper	1	56,995	3,583	386
	moshi / retrofit	1	54,987	1,135	208
	jackson-core / mockserver	1	100,567	5,642	1,143
	antlr4-runtime / fizzed-rocker	1	84,232	128,965	9,372
	httpcomponents-client / wasabi	1	186,329	217	46
	jackson-databind / openAPI-generator	1	387,458	38,813	6,941
	alibaba-druid / dble	1	445,216	53,957	927
	Average	**-**	**171,984**	**31,742**	**2,449**

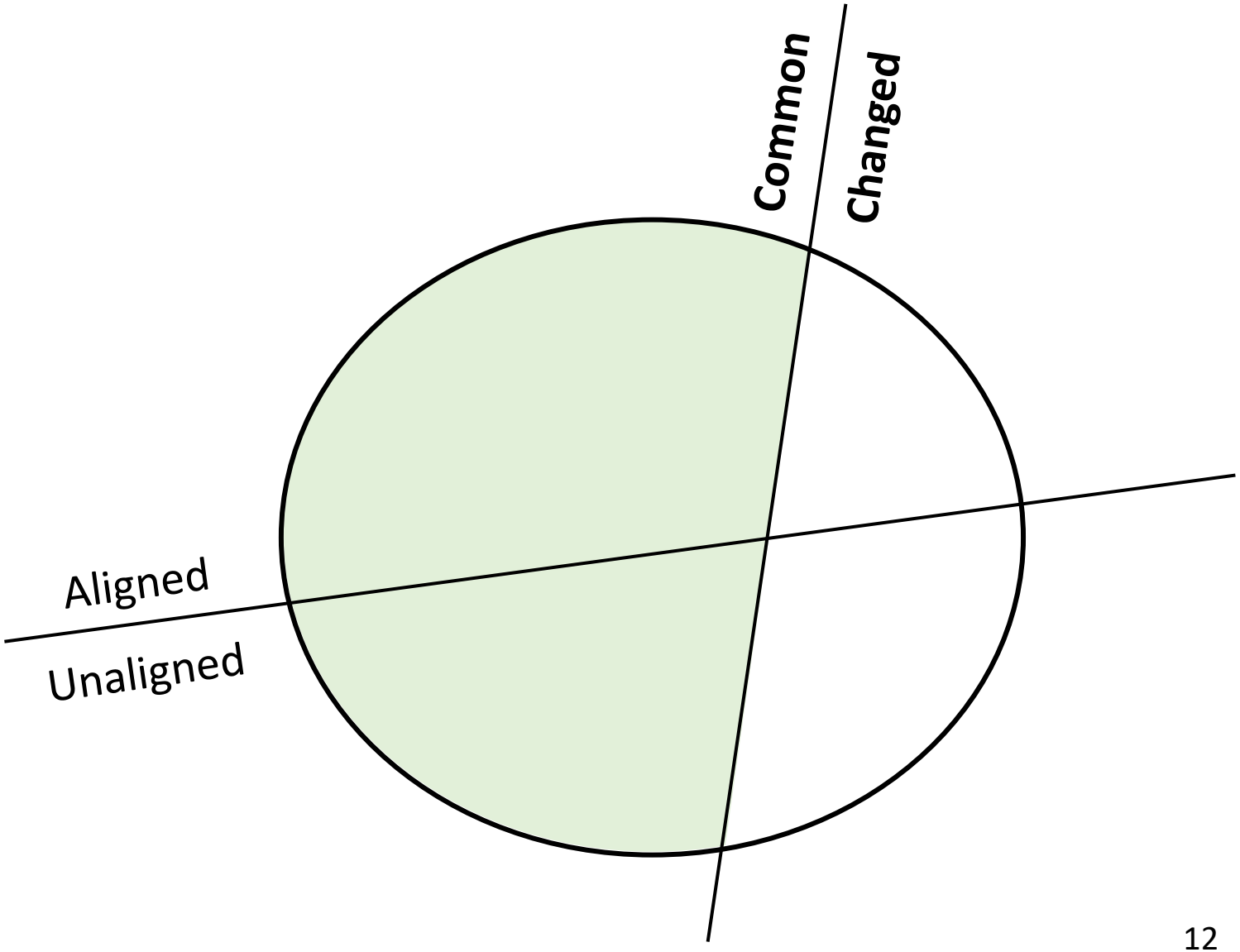
278 Defects4J and **8 large client-library** regression pairs

Slices are > **1.5K** statements, on average.
Not all statements are “interesting”.

Analyzing Slices



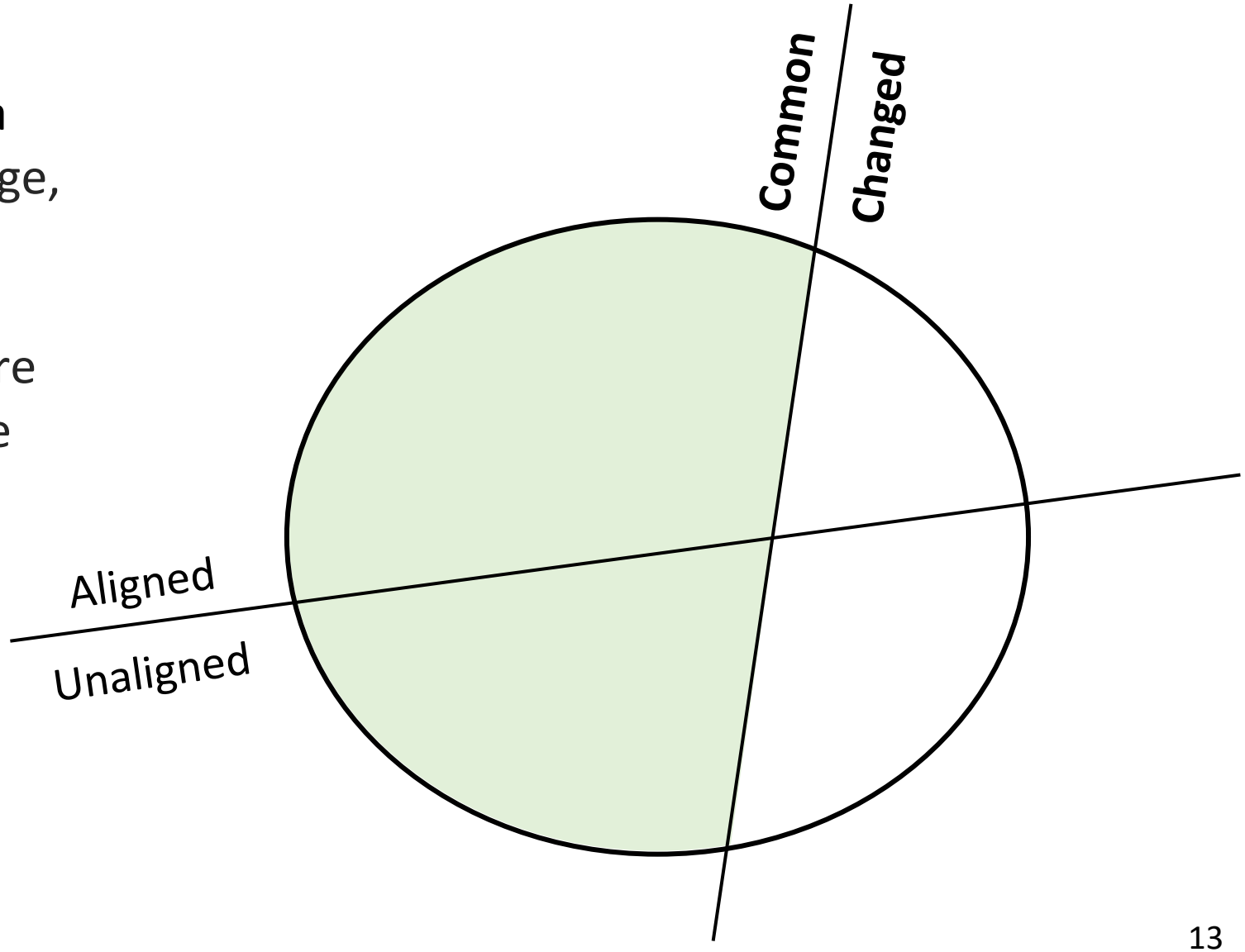
Analyzing Slices



Analyzing Slices

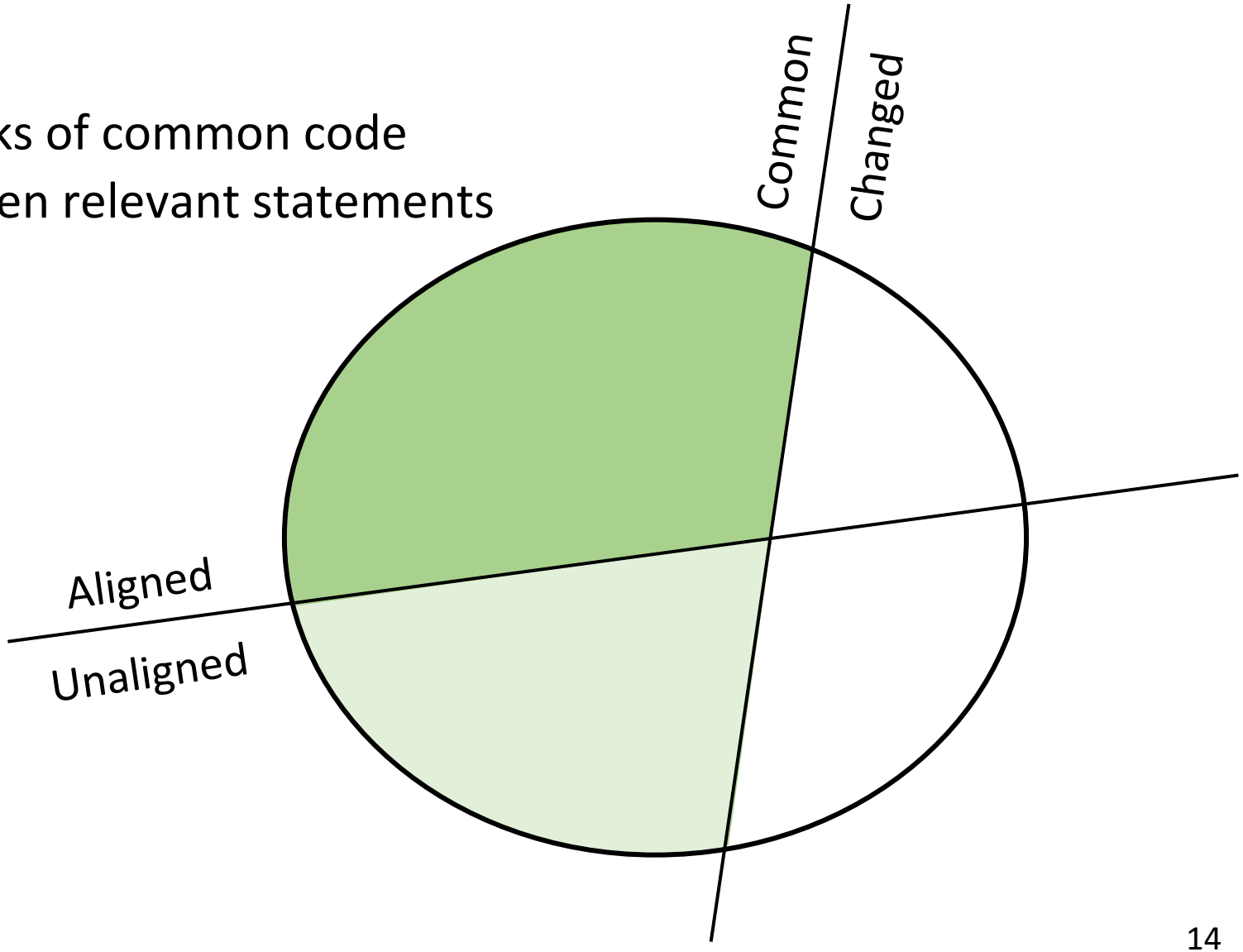
- Large
  (sequences of common code in a slice are **96** statements on average, max **1,032** statements)

- Not directly relevant to the failure
  (but **propagate** the effects of the change through the slice)

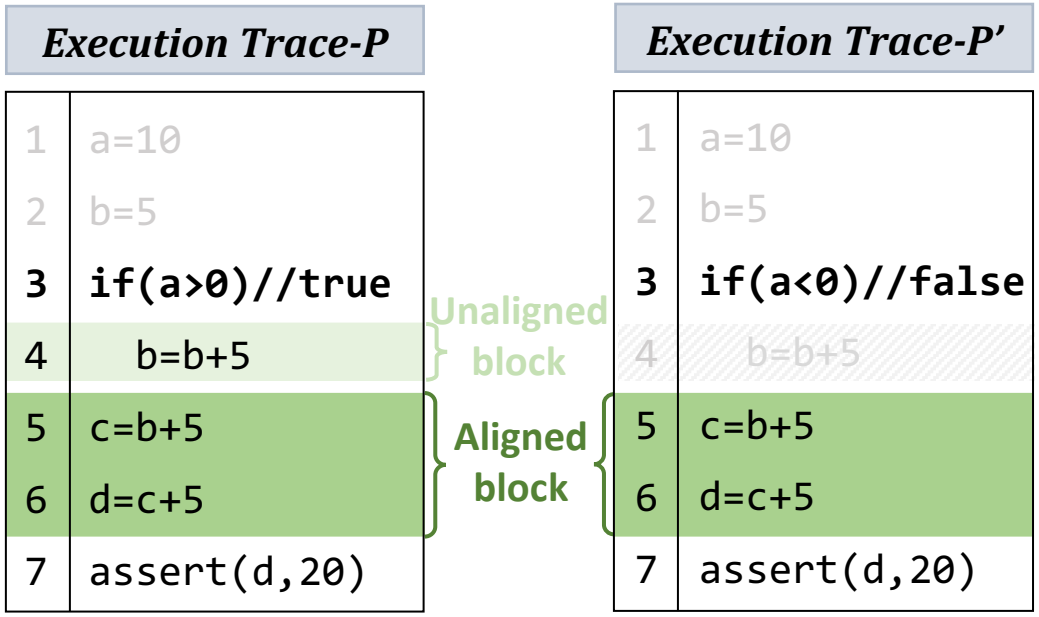


Information-Preserving Slice Summarization (InPreSS) Main Idea

Summarize corresponding blocks of common code while keeping dependencies between relevant statements



InPreSS: Block Extraction and Summarization

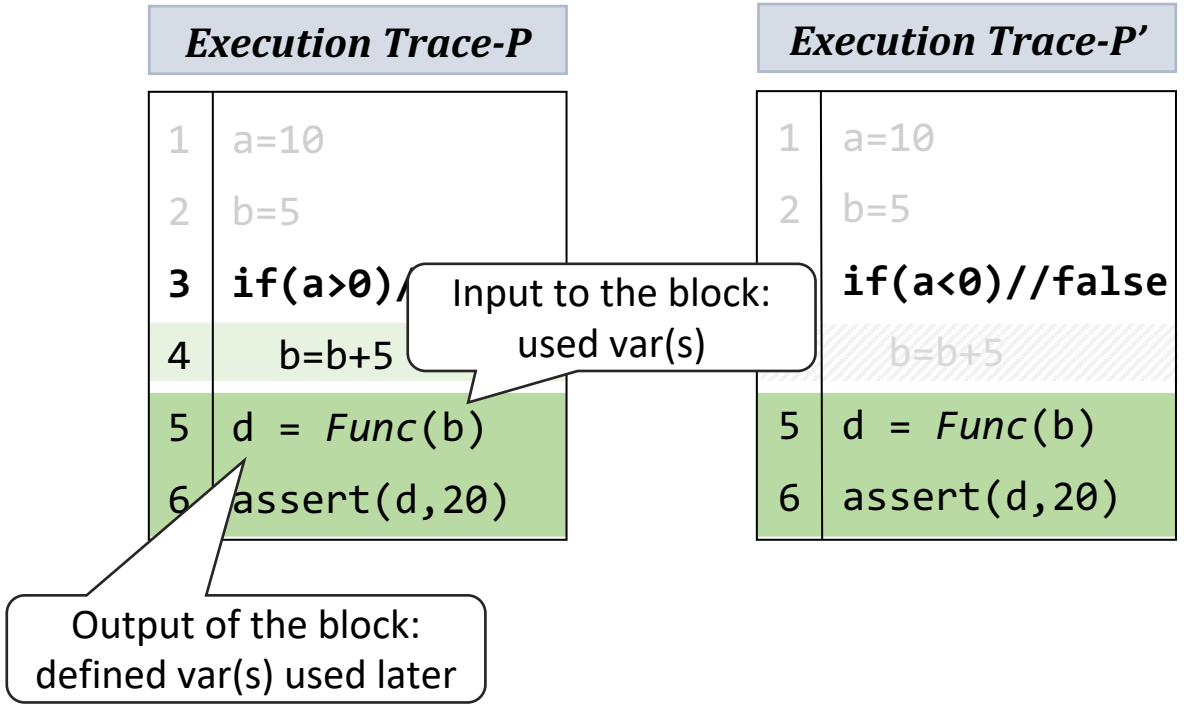


Step 1: separate common blocks into aligned and unaligned

◇ Executed in **one** version only
=> express control divergency as an effect of the change

▢ Executed in **both** versions, with different data
=> express data divergency as an effect of the change

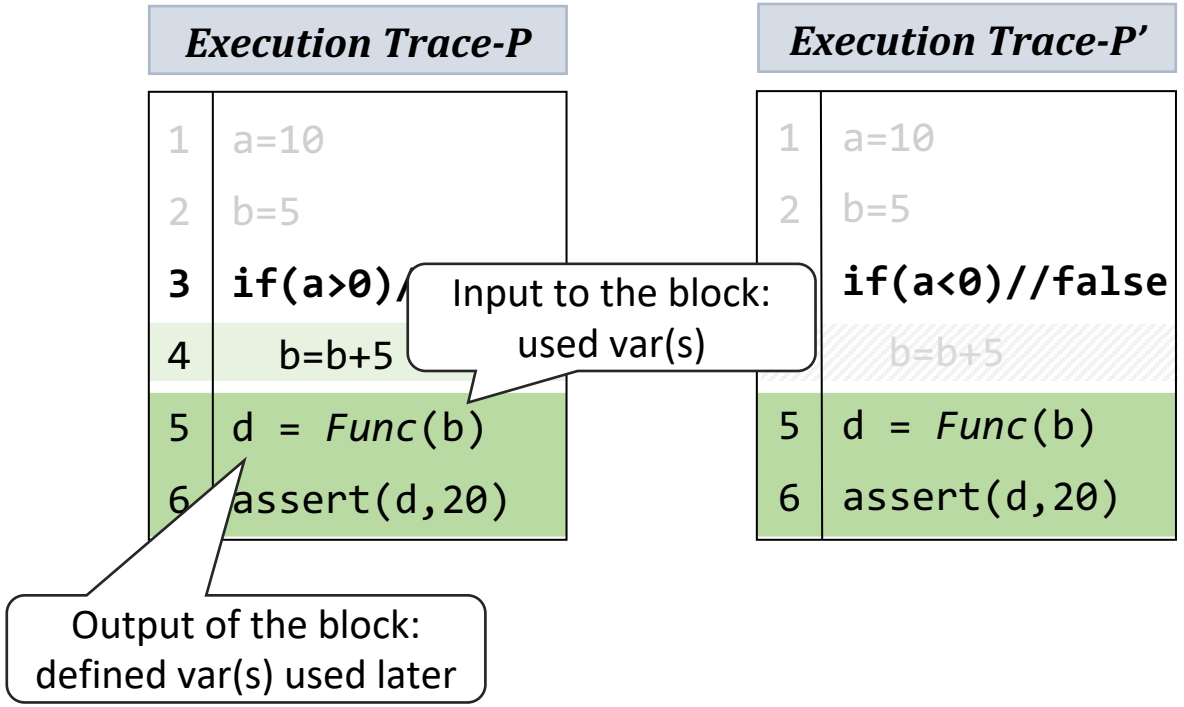
InPreSS: Block Extraction and Summarization



Step 1: separate common blocks into aligned and unaligned

Step 2: summarize dependencies as input-output relationships

InPreSS: Block Extraction and Summarization



Step 1: separate common blocks into aligned and unaligned

Step 2: summarize dependencies as input-output relationships

Produces 76% shorter slices (207 statements, on average)
while keeping the dependencies between the variables in the slice

Summary So Far: Main Contributions

1. An annotated benchmark of regression failures in large and popular client-library pairs

2. A study on the applicability of slicing for troubleshooting regression failures

3. An approach for reducing the size of slices by abstracting contextual information, while preserving its effects on the failure through propagation

Implications and Future Work

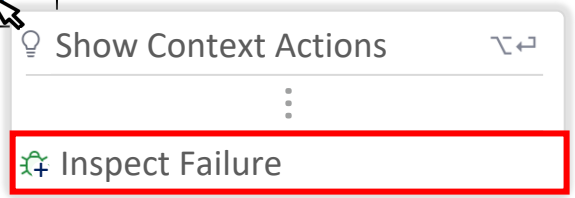
Program-P

```
1  a=10
2  b=5
3  if(a>0)//true
4     b=b+5
5  c=b+5
6  d=c+5
7  assert(d,20)
```

Program-P'

```
1  a=10
2  b=5
3  if(a<0)//false
4     b=b+5
5  c=b+5
6  d=c+5
7  assert(d,20)
```

Promote efficient integration of slicing-based techniques in debugging



Implications and Future Work

Program-P

```
1  a=10
2  b=5
3  if(a>0)//true
4     b=b+5
5  c=b+5
6  d=c+5
7  assert(d,20)
```

Program-P'

```
1  a=10
2  b=5
3  if(a<0)//false
4     b=b+5
5  c=b+5
6  d=c+5
7  assert(d,20)
```

Promote efficient integration of slicing-based techniques in debugging

Irrelevant statements greyed out (including common aligned statements)

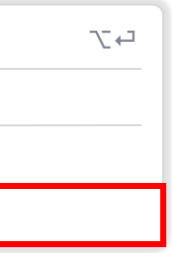
Implications and Future Work

Program-P

```
1  a=10
2  b=5
3  if(a>0)//true
4     b=b+5
5  c=b+5
6  d=c+5
7  assert(d,20)
```

Program-P'

```
1  a=10
2  b=5
3  if(a<0)//false
4     b=b+5
5  c=b+5
6  d=c+5
7  assert(d,20)
```



Promote efficient integration of slicing-based techniques in debugging

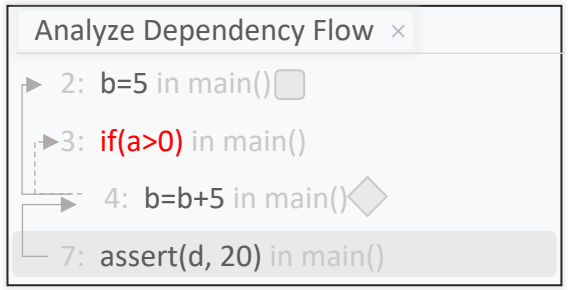
Implications and Future Work

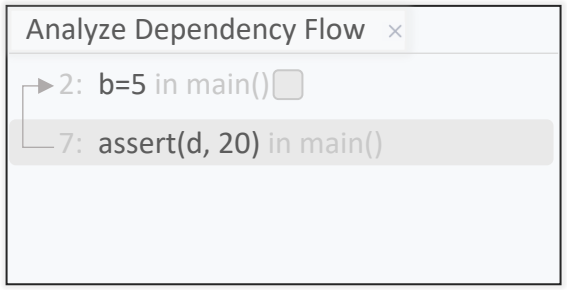
Program-P

```
1  a=10
2  b=5
3  if(a>0)//true
4     b=b+5
5  c=b+5
6  d=c+5
7  assert(d,20)
```

Program-P'

```
1  a=10
2  b=5
3  if(a<0)//false
4     b=b+5
5  c=b+5
6  d=c+5
7  assert(d,20)
```





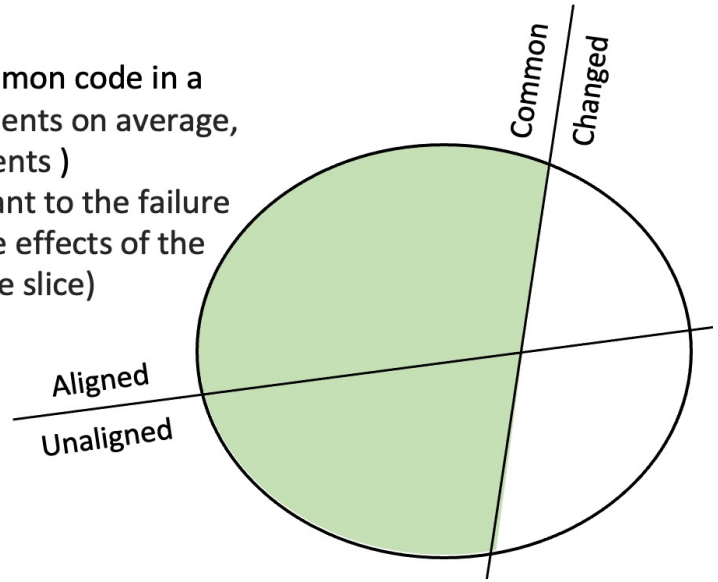
Promote efficient integration of slicing-based techniques in debugging

Summary

The problem: slicing-based techniques are effective to help troubleshooting failures
… but slices are still long and unfocused

Study on the usefulness and properties of slices

- Large
  (sequences of common code in a slice are **96** statements on average, max **1,032** statements)
- Not directly relevant to the failure
  (but **propagate** the effects of the change through the slice)



Context-preserving summarization **technique**

Input to the block:
used var(s)

d = $Func$(b)

Output of the block:
defined var(s) used later

Produces 76% shorter slices (191 statements, on average)
while keeping the dependencies between the variables in the slice

LibRench benchmark: https://zenodo.org/record/7683853
InPreSS and Dual Slicing toolset: https://zenodo.org/record/7684134#.ZF1foS8r1hE