

Metamorphic Coverage

Jinsheng Ba¹, Yuancheng Jiang², Manuel Rigger²

The Chinese University of Hong Kong, Shenzhen¹, National University of Singapore²



香港中文大學(深圳)

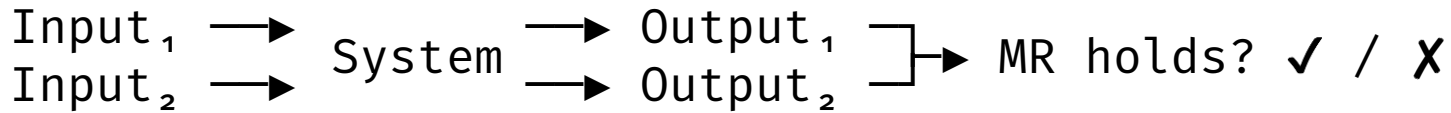
The Chinese University of Hong Kong, Shenzhen



NUS
National University
of Singapore

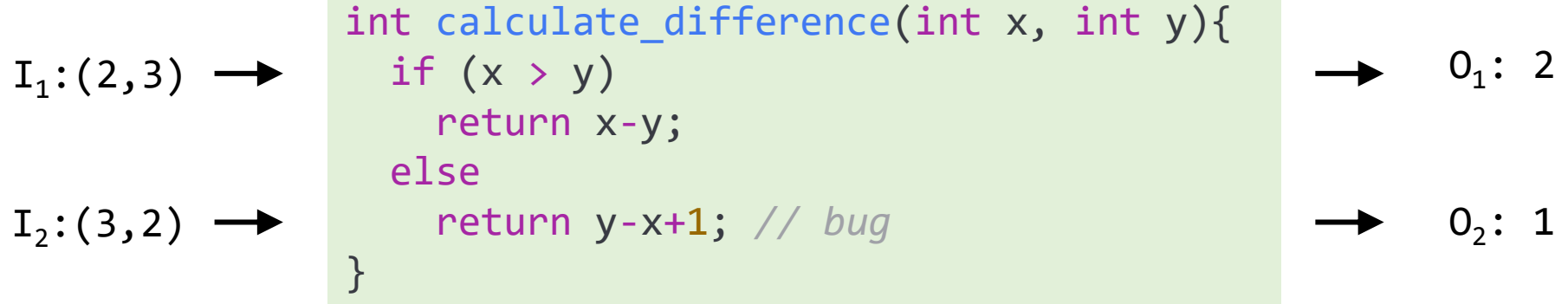
Metamorphic Testing

- A **property-based testing technique** that verifies software using **metamorphic relations (MRs)** instead of explicit expected outputs



Metamorphic Testing Example

MR: if $I_1 = (x, y)$ and $I_2 = (y, x)$, $O_1 = O_2$



Method	Target	Publication
RTI	AI	ICSE'20
MetaOD	AI	ASE'20
EMI	Compiler	PLDI'14
HirGen	Compiler	ISSTA'23
NoREC	DBMS	OOPSLA'20
TLP	DBMS	FSE'20
YinYang	SMT	PLDI'20
SAE	SMT	FSE'21
EDEFuzz	Web	ICSE'24
MROP	Web	ICSE'18

Metamorphic testing is widely used, but how to evaluate its effectiveness?



Method	Target	Publication	Bugs	Code Coverage
RTI	AI	ICSE'20	✓	
MetaOD	AI	Posteriori: too time-consuming	✓	
EMI	Compiler		✓	
HirGen	Compiler	ICSE'20	✓	
NoREC	DBMS	OOPSLA'20	✓	
TLP	DBMS	FSE'20	✓	✓
YinYang	SMT	PLDI'20	✓	✓
SAE	SMT	FSE'21	✓	
EDEFuzz	Web	ICSE'24	✓	
MROP	Web	ICSE'18	✓	

Motivating Example

MR₁: if $l_1 = (x, y)$ and $l_2 = (y, x)$, $O_1 = O_2$

(2, 3) (6, 2)

MR₂: for an integer c , if $l_1 = (x, y)$ and $l_2 = (x + c, y + c)$, $O_1 = O_2$

(2, 3) (6, 2)

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;  
4.     else  
5.         return y-x+1; // bug  
6. }
```

Motivating Example

MR₁: if $l_1 = (x, y)$ and $l_2 = (y, x)$, $O_1 = O_2$

(2,3)(3,2) (6,2)(2,6)

MR₂: for an integer c , if $l_1 = (x, y)$ and $l_2 = (x + c, y + c)$, $O_1 = O_2$

(2,3)(2+1,3+1) (6,2)(6+1,2+1)

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;  
4.     else  
5.         return y-x+1; // bug  
6. }
```

Motivating Example

MR₁: if $l_1 = (x, y)$ and $l_2 = (y, x)$, $O_1 = O_2$

(2, 3)(3, 2) (6, 2)(2, 6)

MR₂: for an integer c , if $l_1 = (x, y)$ and $l_2 = (x + c, y + c)$, $O_1 = O_2$

(2, 3)(2+1, 3+1) (6, 2)(6+1, 2+1)

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;  
4.     else  
5.         return y-x+1; // bug  
6. }
```

Motivating Example

MR₁: if $l_1 = (x, y)$ and $l_2 = (y, x)$, $O_1 = O_2$

(2, 3)(3, 2) (6, 2)(2, 6)

MR₂: for an integer c , if $l_1 = (x, y)$ and $l_2 = (x + c, y + c)$, $O_1 = O_2$

(2, 3)(2+1, 3+1) (6, 2)(6+1, 2+1)

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;  
4.     else  
5.         return y-x+1; // bug  
6. }
```

Motivating Example

MR₁: if $l_1 = (x, y)$ and $l_2 = (y, x)$, $O_1 = O_2$

Line coverage: lines 2-5

(2, 3)(3, 2) (6, 2)(2, 6)

MR₂: for an integer c , if $l_1 = (x, y)$ and $l_2 = (x + c, y + c)$, $O_1 = O_2$

(2, 3)(2+1, 3+1) (6, 2)(6+1, 2+1)

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;  
4.     else  
5.         return y-x+1; // bug  
6. }
```

Motivating Example

MR₁: if $l_1 = (x, y)$ and $l_2 = (y, x)$, $O_1 = O_2$

Line coverage: lines 2-5

(2, 3)(3, 2) (6, 2)(2, 6)

MR₂: for an integer c , if $l_1 = (x, y)$ and $l_2 = (x + c, y + c)$, $O_1 = O_2$

(2, 3)(2+1, 3+1) (6, 2)(6+1, 2+1)

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;  
4.     else  
5.         return y-x+1; // bug  
6. }
```

Motivating Example

MR₁: if $l_1 = (x, y)$ and $l_2 = (y, x)$, $O_1 = O_2$

Line coverage: lines 2-5

(2, 3)(3, 2) (6, 2)(2, 6)

MR₂: for an integer c , if $l_1 = (x, y)$ and $l_2 = (x + c, y + c)$, $O_1 = O_2$

(2, 3)(2+1, 3+1) (6, 2)(6+1, 2+1)

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;  
4.     else  
5.         return y-x+1; // bug  
6. }
```

Motivating Example

MR₁: if $l_1 = (x, y)$ and $l_2 = (y, x)$, $O_1 = O_2$

Line coverage: lines 2-5

(2,3)(3,2) (6,2)(2,6)

MR₂: for an integer c , if $l_1 = (x, y)$ and $l_2 = (x + c, y + c)$, $O_1 = O_2$

(2,3)(2+1,3+1) (6,2)(6+1,2+1)

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;  
4.     else  
5.         return y-x+1; // bug  
6. }
```

Motivating Example

MR₁: if $l_1 = (x, y)$ and $l_2 = (y, x)$, $O_1 = O_2$

Line coverage: lines 2-5

(2, 3)(3, 2) (6, 2)(2, 6)

MR₂: for an integer c , if $l_1 = (x, y)$ and $l_2 = (x + c, y + c)$, $O_1 = O_2$

(2, 3)(2+1, 3+1) (6, 2)(6+1, 2+1) **Line coverage: lines 2-5**

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;
```

Code coverage cannot recognize the bug-finding capability of MR₁.

Idea

MR₁: if $I_1 = (x, y)$ and $I_2 = (y, x)$, $O_1 = O_2$

MR₂: for an integer c , if $I_1 = (x, y)$ and $I_2 = (x + c, y + c)$, $O_1 = O_2$

Examine the difference in the code executed by both inputs

```
1. int calculate_difference(int x, int y){  
2.     if (x > y)  
3.         return x-y;  
4.     else  
5.         return y-x+1; // bug  
6. }
```



Coverage for test case t_a : (2,3)

```
1. int calculate_difference(int x, int y) {  
2.   if (x > y) {  
3.     return x - y;  
4.   } else {  
5.     return y - x + 1;  
6.   }  
7. }
```

Coverage for test case t_b : (3,2)

```
1. int calculate_difference(int x, int y) {  
2.   if (x > y) {  
3.     return x - y;  
4.   } else {  
5.     return y - x + 1;  
6.   }  
7. }
```



Metamorphic coverage for
a single pair of test cases

Metamorphic Coverage 1

```
1. int calculate_difference(int x, int y) {  
2.   if (x > y) {  
3.     return x - y;  
4.   } else {  
5.     return y - x + 1;  
6.   }  
7. }
```

Metamorphic Coverage 2

```
1. int calculate_difference(int x, int y) {  
2.   if (x > y) {  
3.     return x - y;  
4.   } else {  
5.     return y - x + 1;  
6.   }  
7. }
```

Metamorphic coverage for
a test set



Sum:
Lines 3,5

Method

- Easy to understand
- Lightweight (implemented in around 100 lines of Python code)
- Based on existing coverage tools, such as gcov

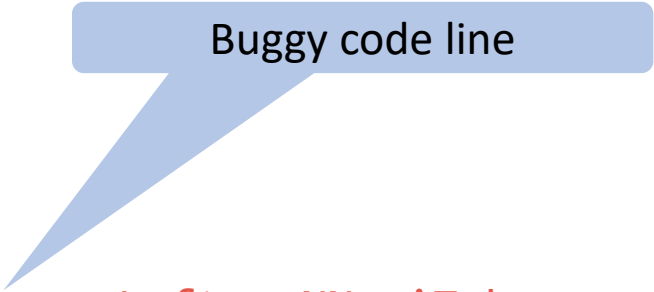
Historical Bugs

- We investigate previously found bugs and their relations to metamorphic coverage.

```
CREATE TABLE t0 (c0) ;  
INSERT INTO t0 (c0) VALUES (NULL) ;  
CREATE INDEX i0 ON t0 (1) WHERE c0 NOT NULL;  
SELECT COUNT (*) FROM t0 WHERE (t0.c0 IS FALSE) IS FALSE; -- {0}  
SELECT SUM(count) FROM (SELECT (t0.c0 IS FALSE) IS FALSE as count  
FROM t0) asasdf; -- {1}
```

Historical Bugs

```
--- src/expr.c
+++ src/expr.c
-5034,11 +5034,11
  switch( p->op ){
...
  case TK_TRUTH: {
    if( seenNot ) return 0;
    if( p->op2!=TK_IS ) return 0;
-   return exprImpliesNotNull(pParse, p->pLeft, pNN, iTab, seenNot);
+   return exprImpliesNotNull(pParse, p->pLeft, pNN, iTab, 1);
```



Buggy code line

<https://www.sqlite.org/src/info/45ff2b1f2693bb02>

Historical Bugs

```
--- src/expr.c  
+++ src/expr.c  
-5034,11 +5034,11  
switch( p->op ){
```

```
...
```

```
case TK_TRUTH: {  
    if( seenNot ) return 0;  
    if( p->op2!=TK_IS ) return 0;  
- return exprImpliesNotNull(pParse, p->pLeft, pNN, iTab, seenNot);  
+ return exprImpliesNotNull(pParse, p->pLeft, pNN, iTab, 1);
```

Metamorphic coverage

Buggy code line

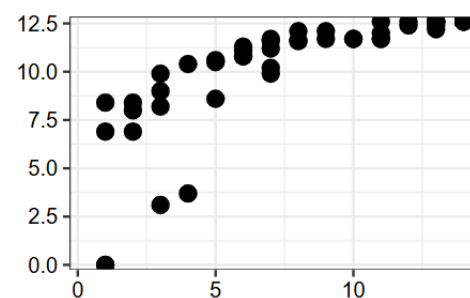
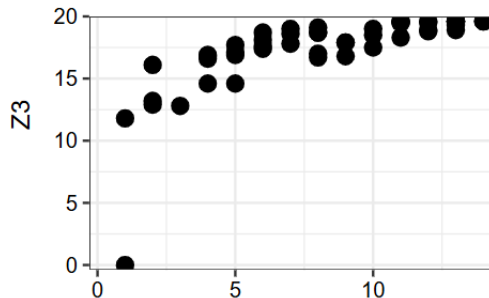
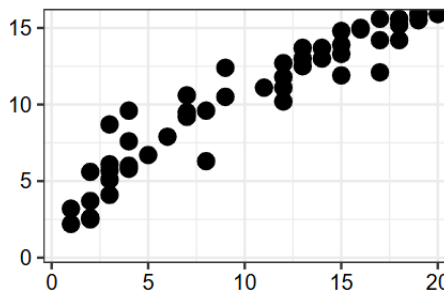
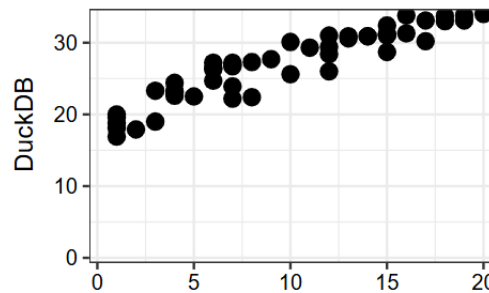
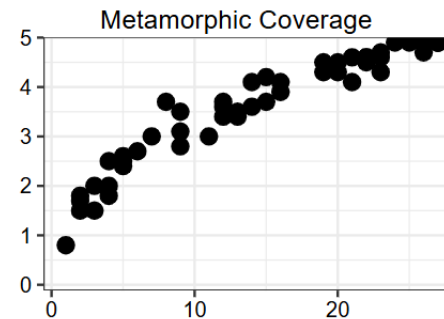
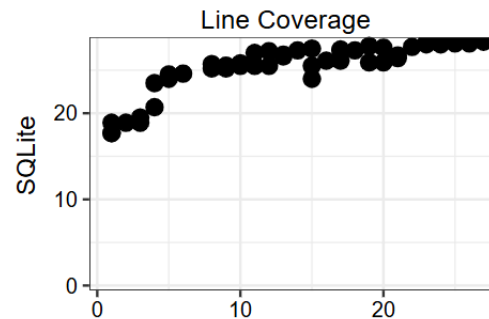
<https://www.sqlite.org/src/info/45ff2b1f2693bb02>

Evaluation

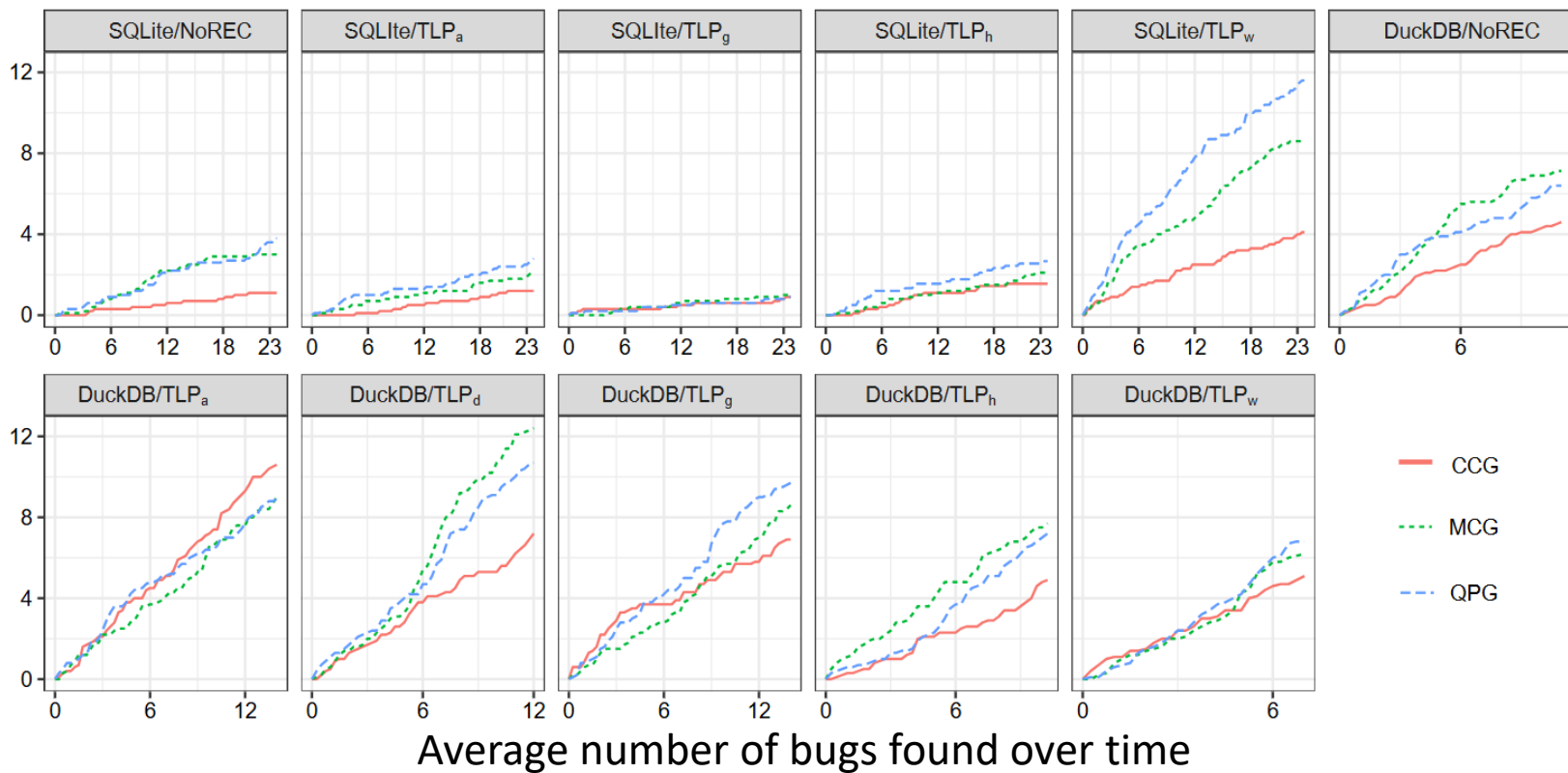
Program	All	Overlapping	Non-overlapping
SQLite	27	18	9
DuckDB	20	15	5
TVM	2	2	0
Z3	14	14	0
CVC4	1	1	0
Sum:	64	50	14

Within 50/64 historical bugs, metamorphic coverage overlaps buggy code lines

Line coverage and metamorphic coverage against the number of bugs



Number of bugs



Using metamorphic coverage as guidance
improves bug-finding efficiency by 41%

Program	Method	Line Coverage	Metamorphic Coverage
SQLite	TLPw	21.84%	1.96%
	TLPg	22.49%	1.87%
	TLPh	22.61%	1.42%
	TLPd	22.14%	2.14%
	TLPa	22.79%	2.56%
	NoREC	22.68%	2.94%
DuckDB	TLPw	20.45%	3.37%
	TLPg	21.47%	3.82%
	TLPh	21.17%	3.45%
	TLPd	21.29%	3.52%
	TLPa	20.99%	4.31%
	NoREC	20.35%	5.97%
TVM	HirGenmut	15.30%	3.27%
	HirGenopt	17.53%	5.08%
Z3	YinYangsat	14.89%	2.84%
	YinYangunsat	16.49%	4.13%
	SAE	14.80%	3.75%
CVC4	YinYangsat	22.89%	2.85%
	YinYangunsat	25.21%	1.86%
	SAE	19.67%	2.17%

Average metamorphic coverage is 3.26%

A large room
for
improvement

Automated Discovery of Test Oracles for Database Management Systems Using LLMs

QIUYANG MANG*, UC Berkeley, USA

RUNYUAN HE, UC Berkeley, USA

SUYANG ZHONG, National University of S

XIAOXUAN LIU, UC Berkeley, USA

HUANCHEN ZHANG, Tsinghua University

ALVIN CHEUNG, UC Berkeley, USA

Approach	Metamorphic Coverage
SQLancer	3.26%
Argus	17.82%
Improvement	5.473×

Mang, Qiuyang, et al. "Automated Discovery of Test Oracles for Database Management Systems Using LLMs." *Proceedings of the ACM on Management of Data* 4.3 (SIGMOD (2026): 1-28.

Conclusion

Metamorphic Testing Example

MR: if $I_1 = (x, y)$ and $I_2 = (y, x)$, $O_1 = O_2$

$I_1: (2, 3) \rightarrow$ `int calculate_difference(int x, int y){
if (x > y)
return x-y;
else
return y-x+1; // bug
}` $O_1: 2$

$I_2: (3, 2) \rightarrow$ `int calculate_difference(int x, int y){
if (x > y)
return x-y;
else
return y-x+1; // bug
}` $O_2: 1$



3

Method	Target	Publication	Bugs	Code Coverage?
RTI	AI	ICSE'20	✓	
MetaOD	AI	Posteriori: too time-consuming	✓	
EMI	Compile		✓	
HirGen	Compiler		✓	
NoREC	DBMS	OOPSLA'20	✓	
TLP	DBMS	FSE'20	✓	✓
YinYang	SMT	PLDI'20	✓	✓
SAE	SMT	FSE'21	✓	
EDEFuzz	Web	ICSE'24	✓	
MROP	Web	ICSE'18	✓	

5

Motivating Example

MR₁: if $I_1 = (x, y)$ and $I_2 = (y, x)$, $O_1 = O_2$ Line coverage: lines 2-5
(2, 3) (3, 2) (6, 2) (2, 6)

MR₂: for an integer c , if $I_1 = (x, y)$ and $I_2 = (x + c, y + c)$, $O_1 = O_2$
(2, 3) (2+1, 3+1) (6, 2) (6+1, 2+1) Line coverage: lines 2-5

```
1. int calculate_difference(int x, int y){
2.   if (x > y)
3.     return x-y;
}
```

Code coverage cannot recognize the bug-finding capability of MR₁.

34

Coverage for test case $t_1: (2, 3)$
1. int calculate_difference(int x, int y) {
2. if (x > y) {
3. return x - y;
4. } else {
5. return y - x + 1;
6. }
7. }

Coverage for test case $t_2: (3, 2)$
1. int calculate_difference(int x, int y) {
2. if (x > y) {
3. return x - y;
4. } else {
5. return y - x + 1;
6. }
7. }

Metamorphic coverage for a single pair of test cases

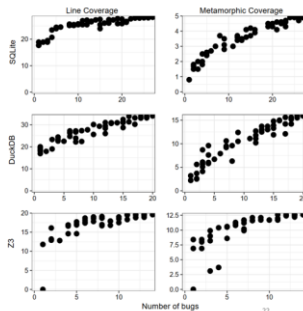
Metamorphic Coverage 1	Metamorphic Coverage 2
1. int calculate_difference(int x, int y) {	1. int calculate_difference(int x, int y) {
2. if (x > y) {	2. if (x > y) {
3. return x - y;	3. return x - y;
4. } else {	4. } else {
5. return y - x + 1;	5. return y - x + 1;
6. }	6. }
7. }	7. }

Sum: Lines 3, 5

Metamorphic coverage for a test set

35

Line coverage and metamorphic coverage against the number of bugs



22

Automated Discovery of Test Oracles for Database Management Systems Using LLMs

QIUYANG MANG¹, UC Berkeley, USA
RUYUAN HE, UC Berkeley, USA
SUYANG ZHONG², National University of Singapore
XIAOXUAN LIU, UC Berkeley, USA
HUANCHEN ZHANG, Tsinghua University
ALVIN CHEUNG, UC Berkeley, USA

Approach	Metamorphic Coverage
SQLancer	3.26%
Argus	17.82%
Improvement	5.473x

Mang, Qiuyang, et al. "Automated Discovery of Test Oracles for Database Management Systems Using LLMs." Proceedings of the ACM on Management of Data 4.3 (SIGMOD 2020): 1-28.

25