

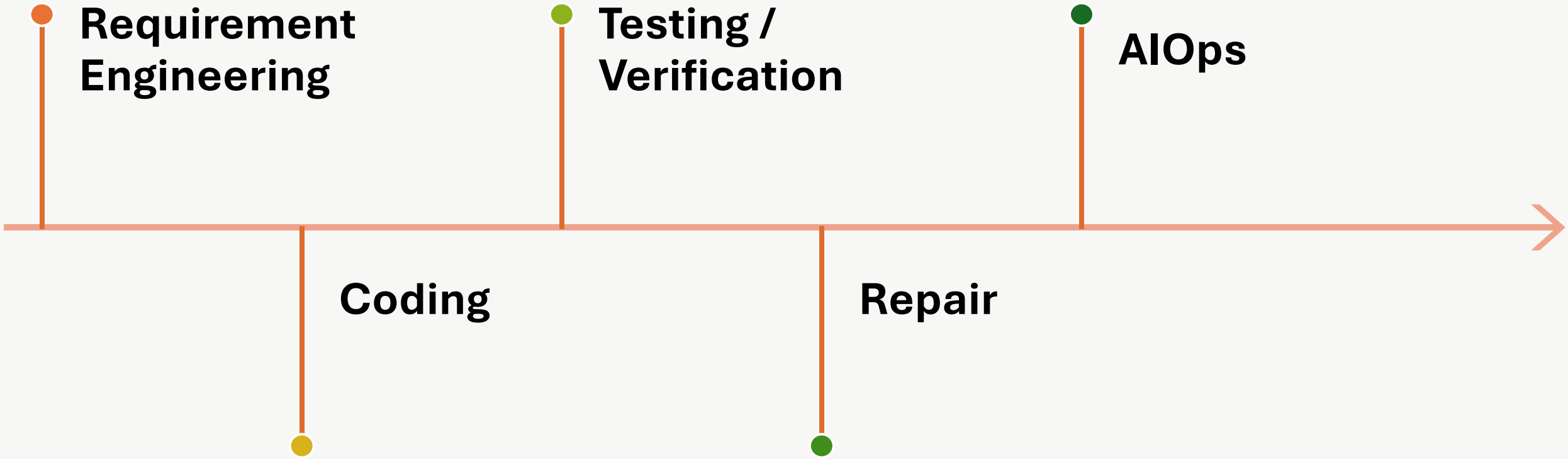
香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

CSC4801
AI-assisted Software Engineering

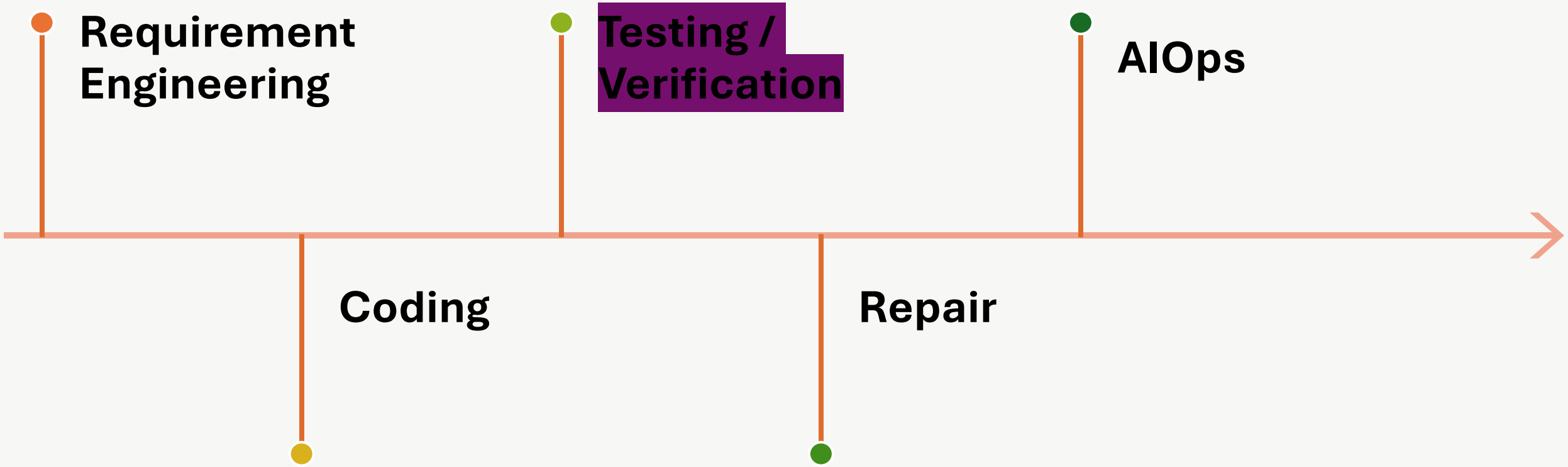
Lecture 12: Testing

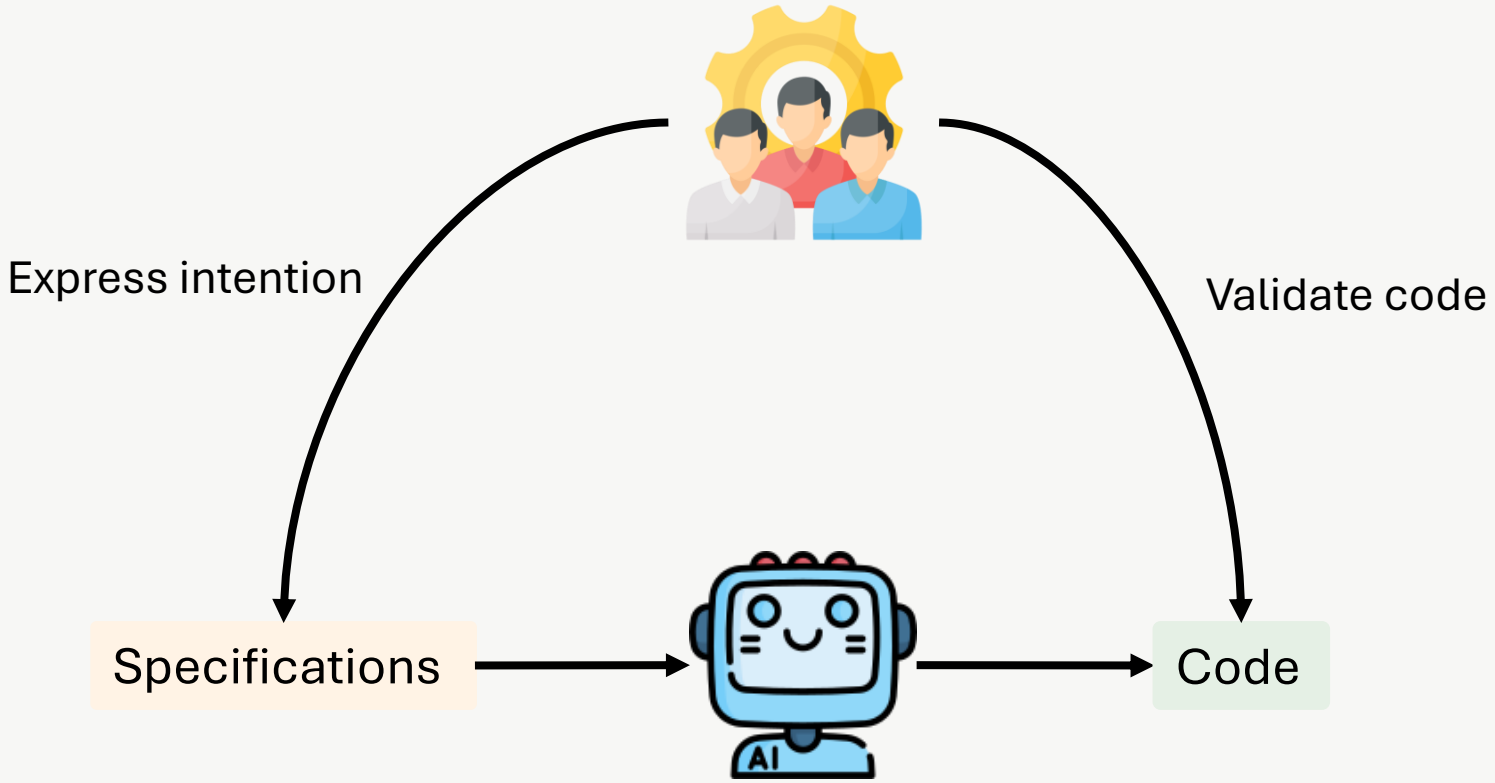
Instructor: Jinsheng Ba | 2026 Fall

AI-Assisted Software Engineering



AI-Assisted Software Engineering





In LLM4SE, the roles of human are intention expression and code validation.

Let’s see how to validate code

Testing

Verification

Testing

Verification

Testing



Testing



Test cases





Code



{ | 1 | } Incorrect result



Unexpected slowdown



Crash

Test oracles

Problem: How to utilize LLMs to efficiently find bugs?

Core Challenges For Automatic Testing

Generating Test Cases →  → Validating Results

- Test oracle (bug identification)
  - Logic bugs: How to know the result is incorrect?
  - Performance issues: How to know an execution time is unexpected?



{ **0** }

1s? 5s?

- Test case generation
  - How to automatically explore huge states of target systems?



Test Case Generation

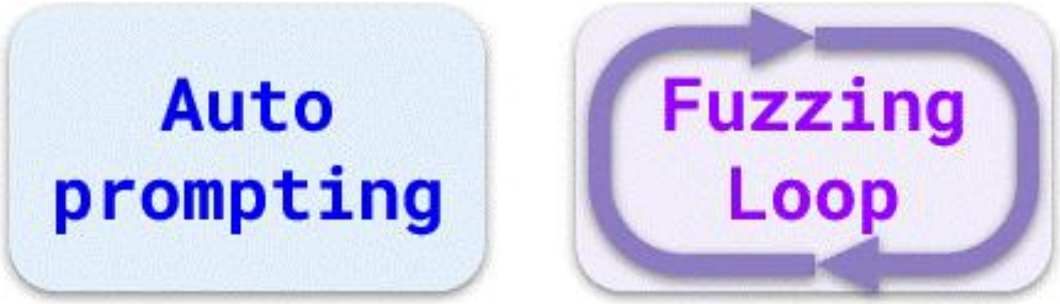
Use LLMs to directly generate inputs for all kinds of programs



https://fuzz4all.github.io/

Prompting Inputs

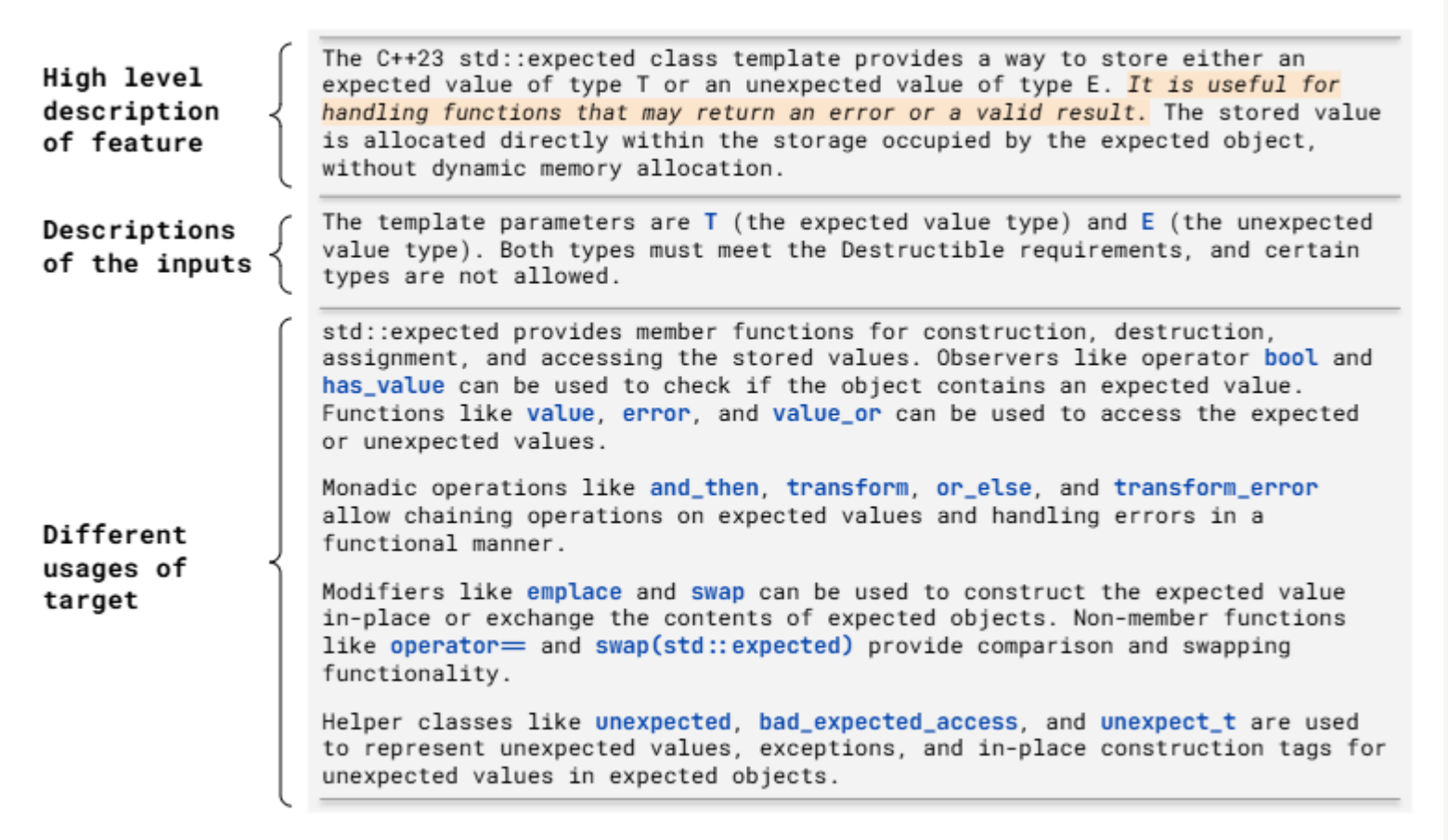
Use LLMs to directly generate inputs for all kinds of programs



Fuzz4All: https://fuzz4all.github.io/

Prompting Inputs

A kind of prompt engineering



https://fuzz4all.github.io/

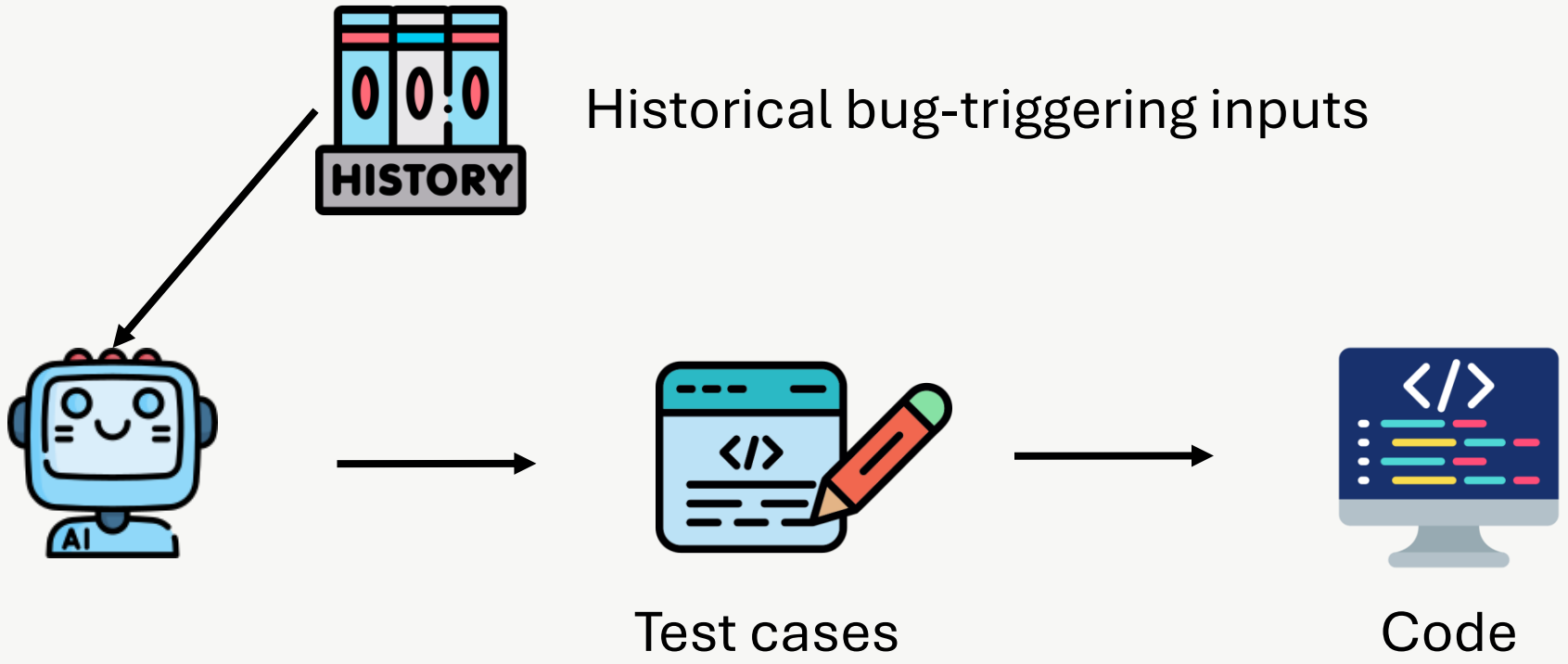
Prompting Inputs

Effective results across different programs.



https://fuzz4all.github.io/

Using Historical Bug-triggering Inputs



FuzzGPT: https://arxiv.org/pdf/2304.02014

Using Historical Bug-triggering Inputs

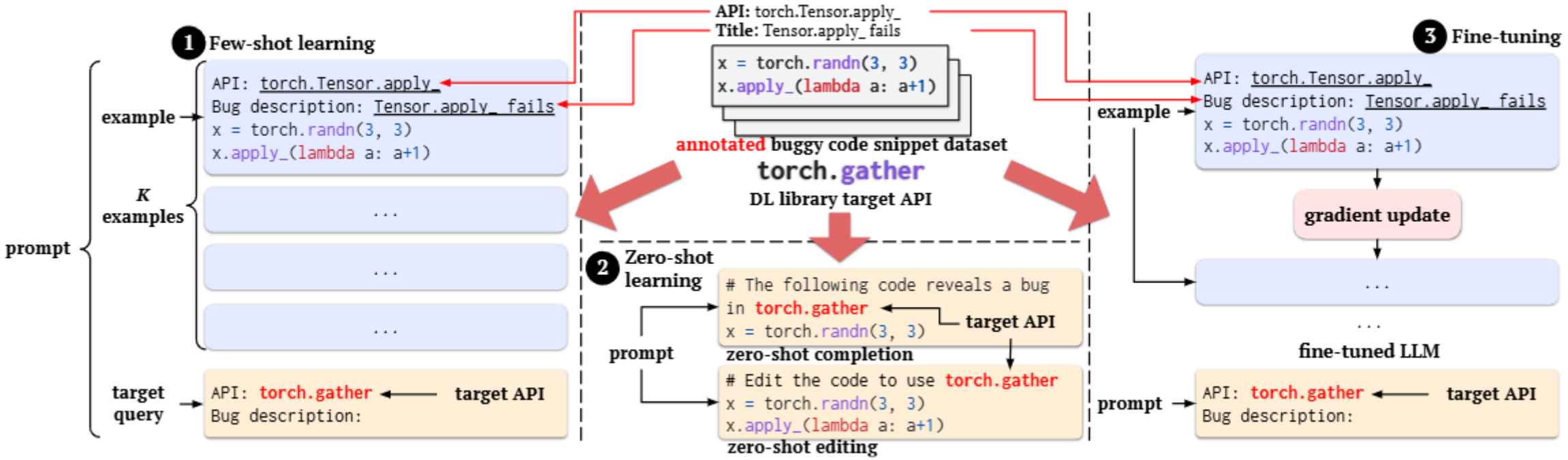
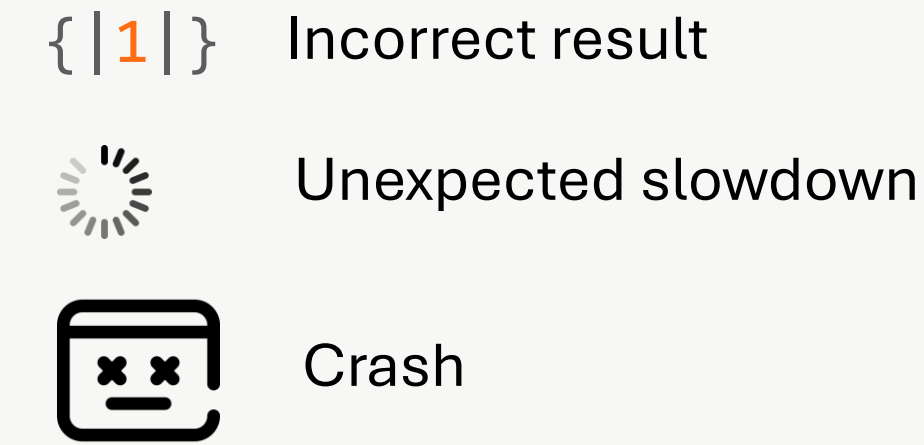


Figure 4: Fine-tuning, zero-shot, and few-shot learning for DL Library Fuzzing.

FuzzGPT: https://arxiv.org/pdf/2304.02014

Test Oracles



Test oracles

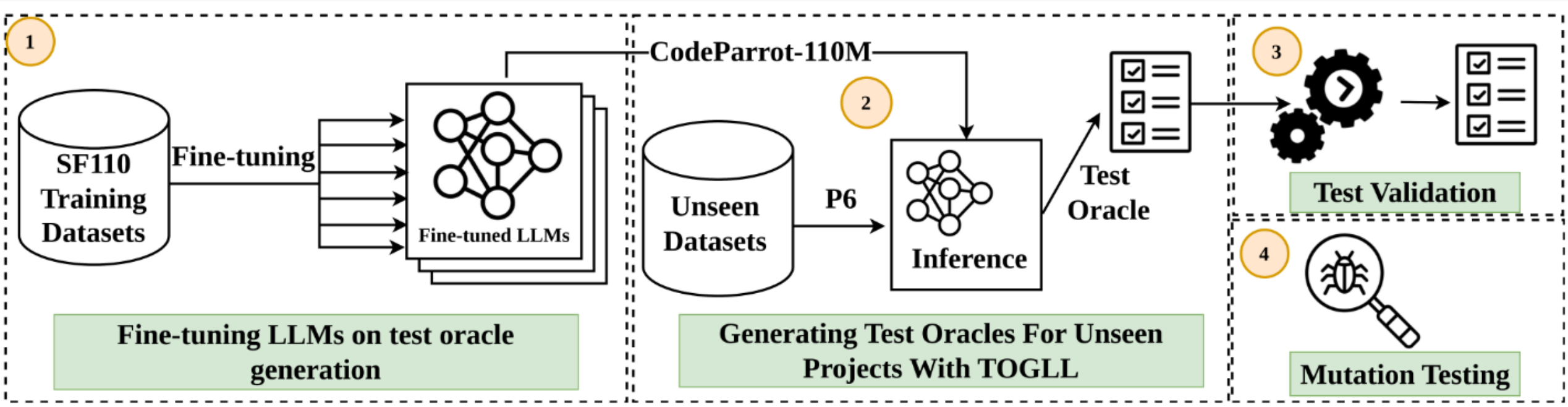
Generating Assertions

```
public void testPushAndPopStack() {

     Stack<Integer> stack = new Stack<>();
     stack.push(10);
     Integer val = stack.pop();
     assertEquals(Integer.valueOf(10), val);
}
```

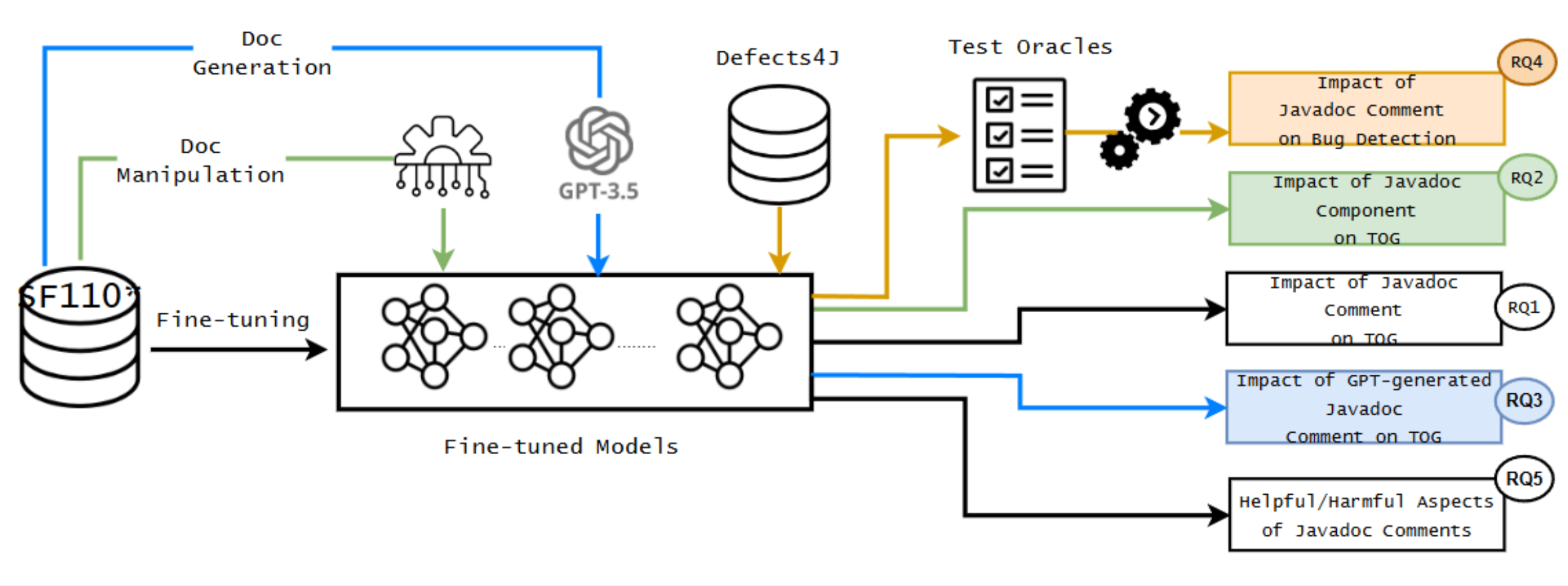
https://arxiv.org/pdf/2405.03786v1

Fine-Tuning Models



TOGLL: https://arxiv.org/pdf/2405.03786v1

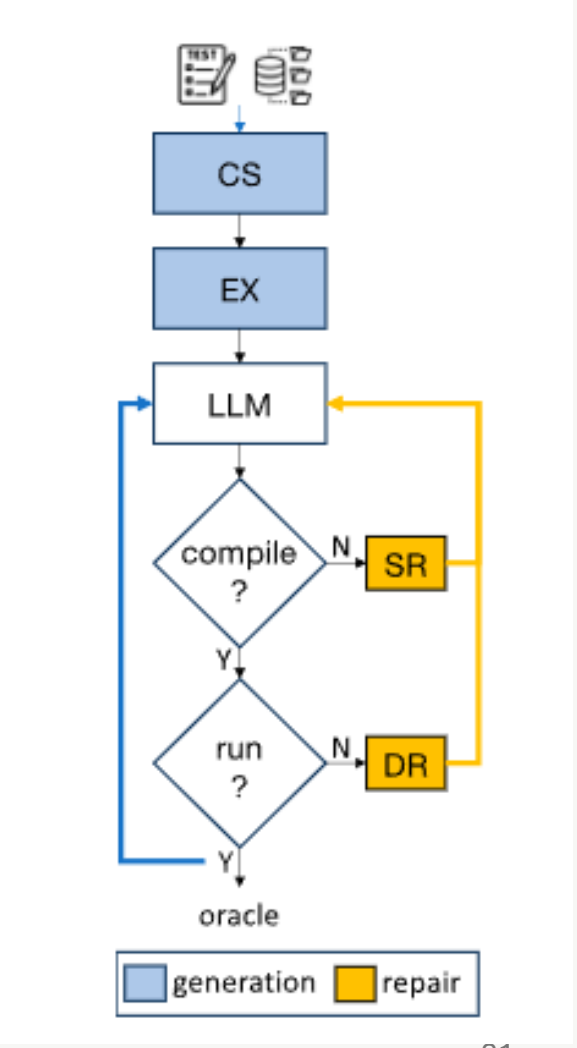
Integrating Doc



Doc2OracLL: https://arxiv.org/pdf/2412.09360

Neural-Symbolic Method: Static Analysis + LLM

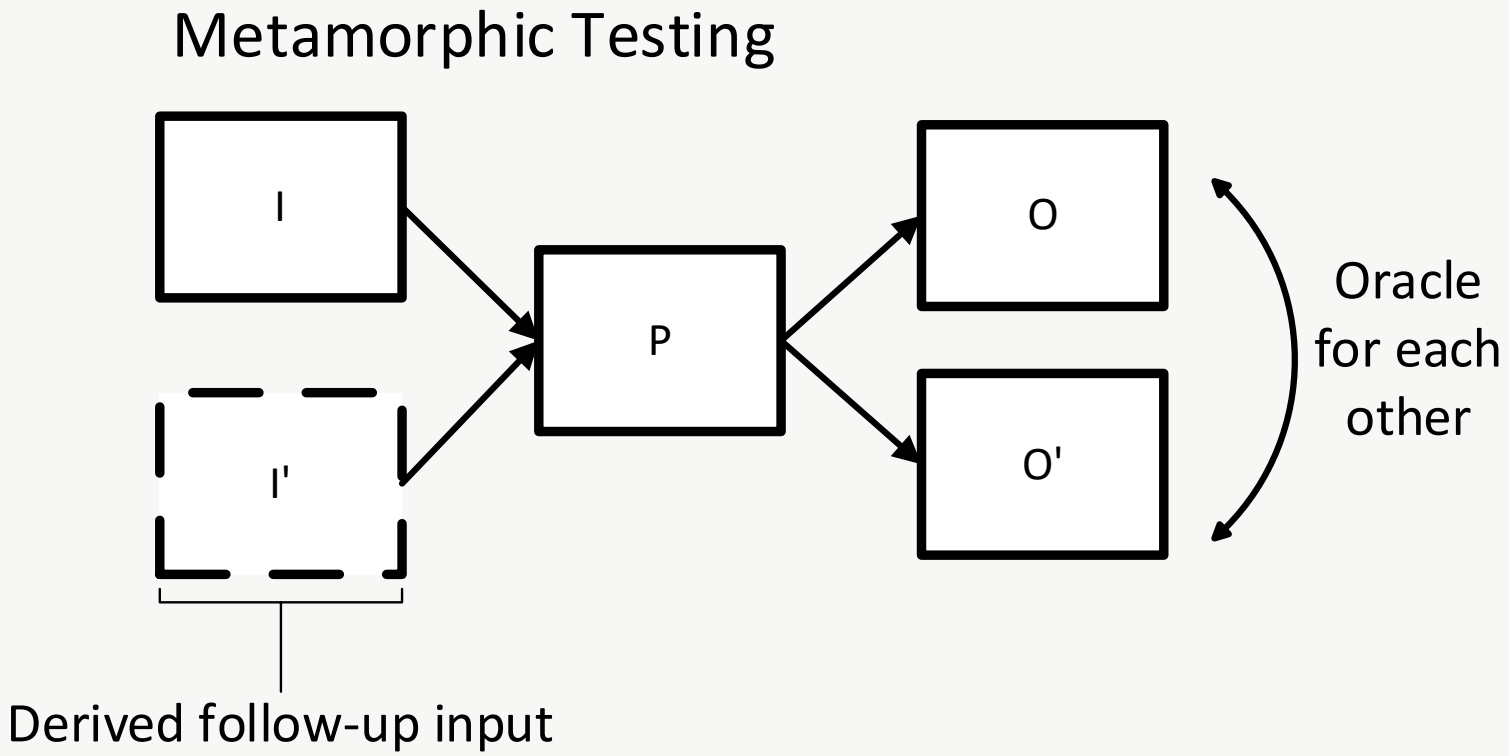
- Test name
- Test setup method
- Test prefix operations
- Using static tracing tools to obtain state change trajectory of the current object



Put these static analysis data into prompts.

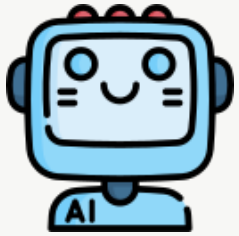
ChatAssert: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10804561

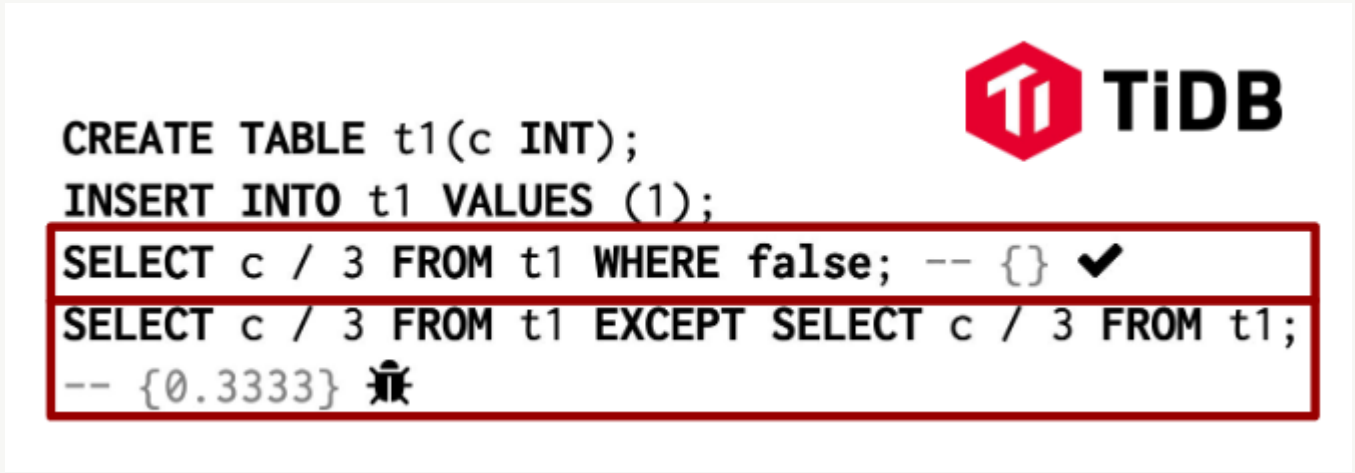
Metamorphic Testing



Database Systems Metamorphic Testing

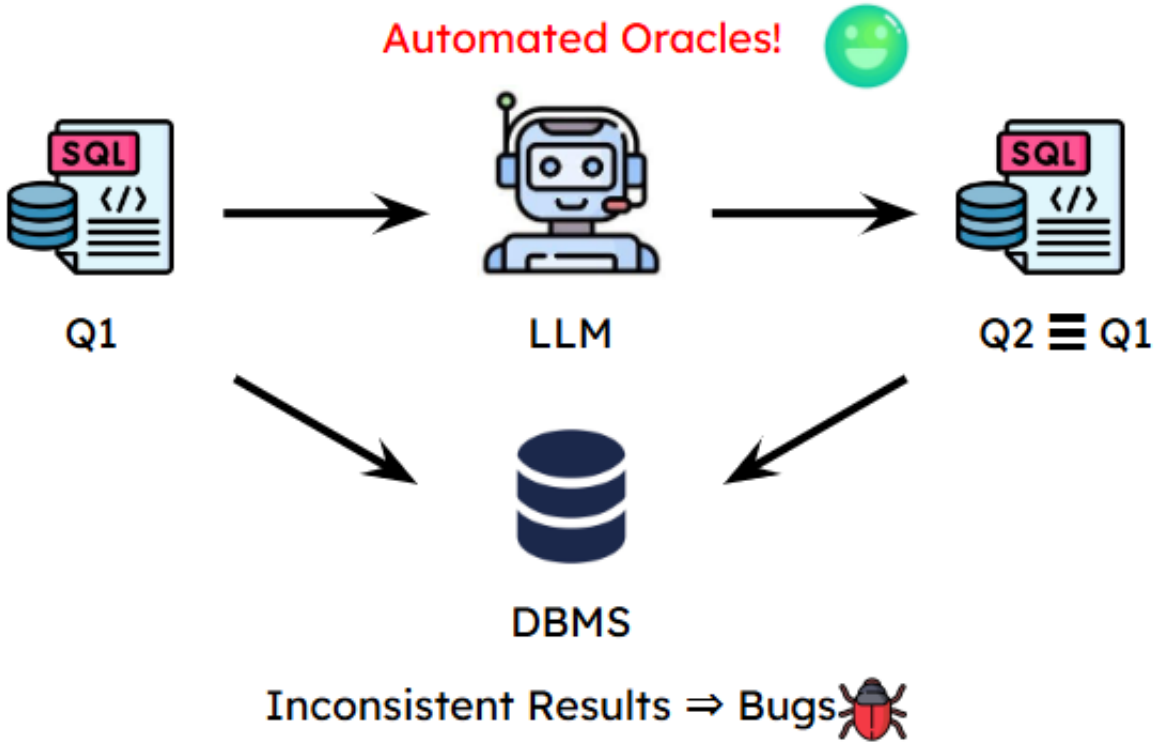
Call LLMs to generate semantically equivalent queries





Argus: https://arxiv.org/pdf/2510.06663

Using LLM to Break the Endless Cycle



C1: LLMs are slow and expensive

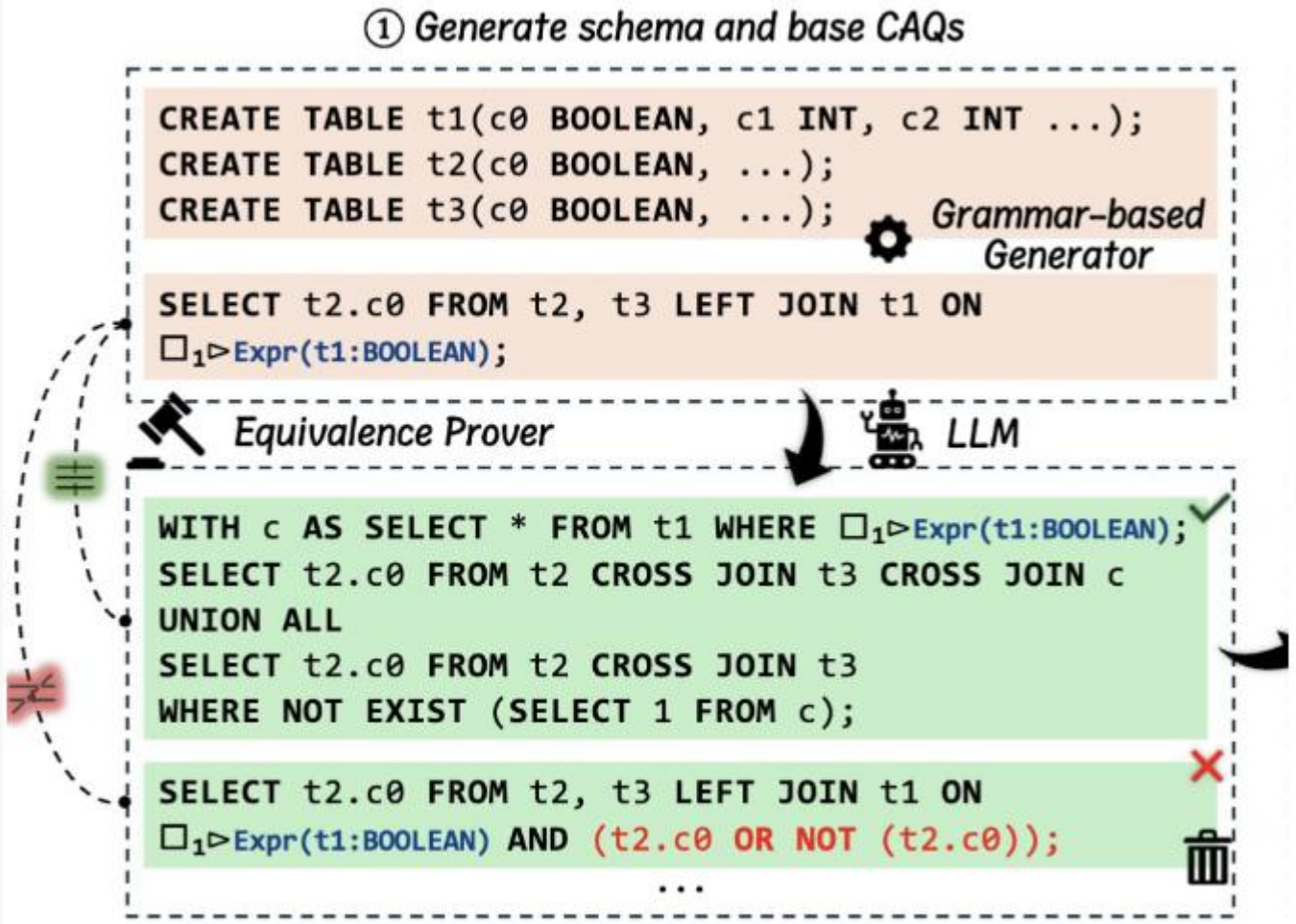
We needs about **100K** test cases to detect one unique bugs in mature DBMS :0

C2: Hallucination ⇒ False Alarms

Filtering true bug reports from a lot of **false positives** is crazy for developers

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.

M2: Using Verification to Avoid Inequivalent CAQs



SQL Equivalence Decider can conservatively prove the equivalence between a pair of SQL queries.

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.

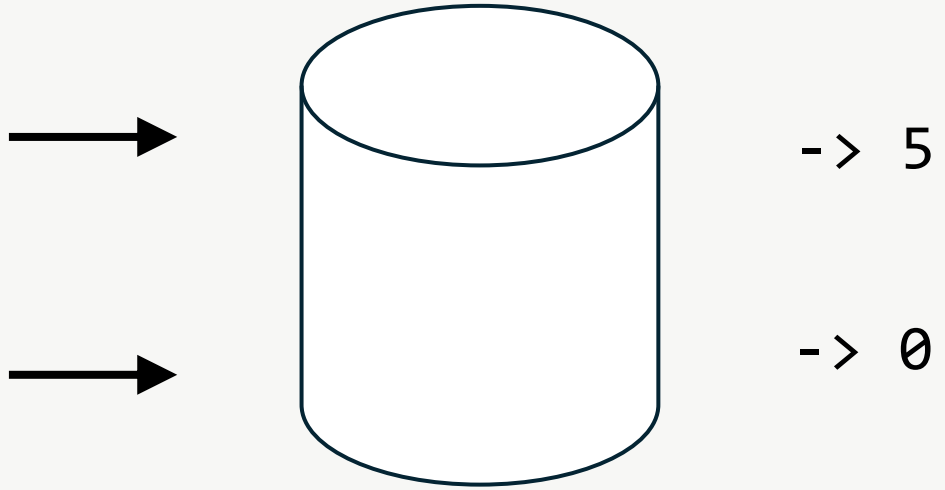
Metamorphic Testing

- Applicable to specific inputs
- Unaware of the internal execution
- Lack of systematic validation

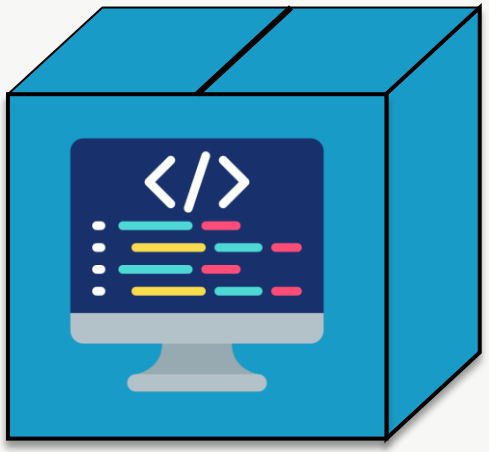
```
SELECT MAX(c0) FROM t0
```

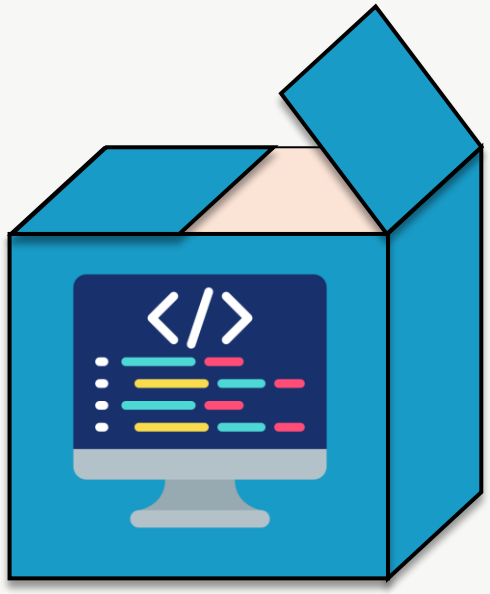
=

```
SELECT MAX(c0) FROM t0
    WHERE TRUE
```



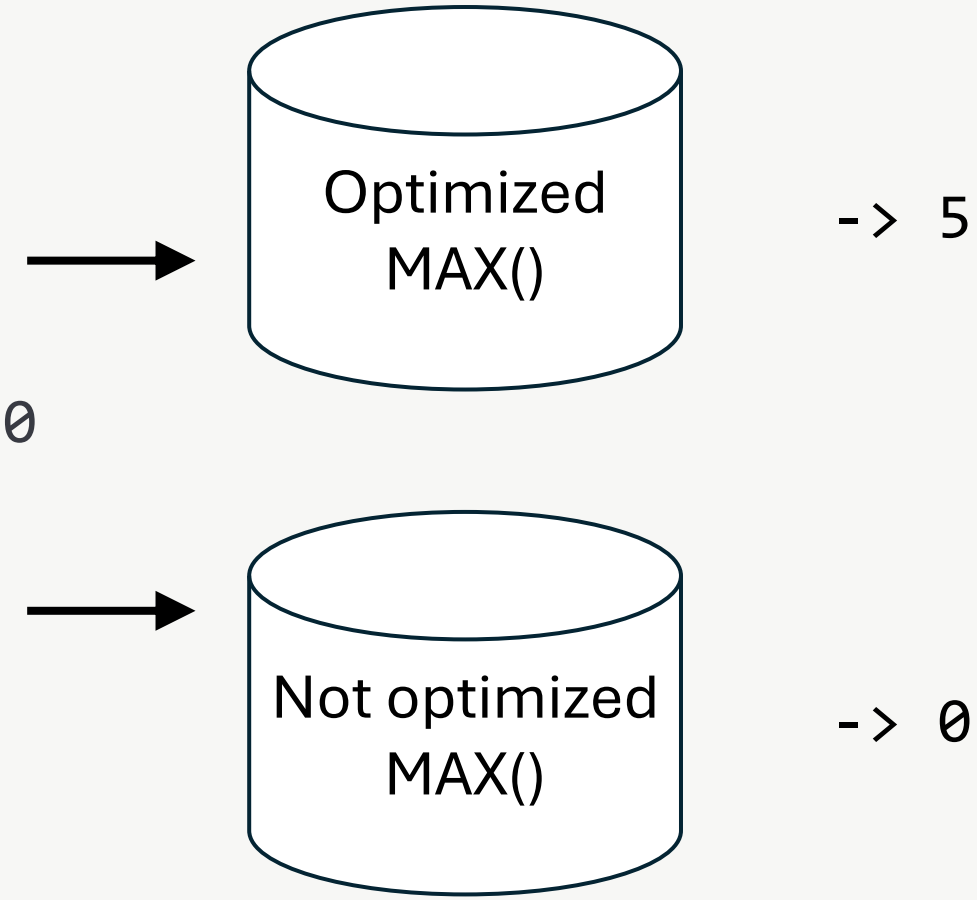




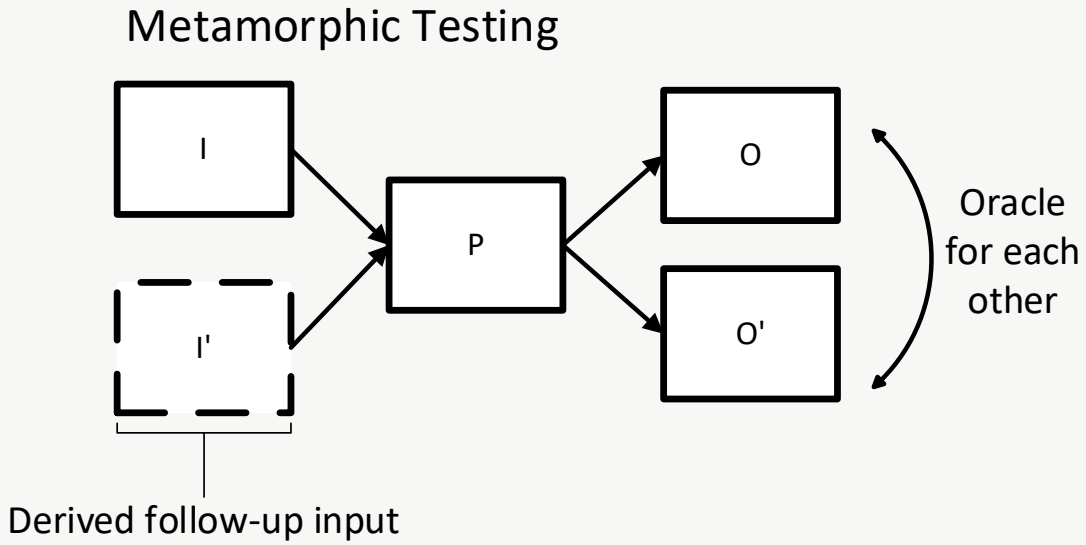


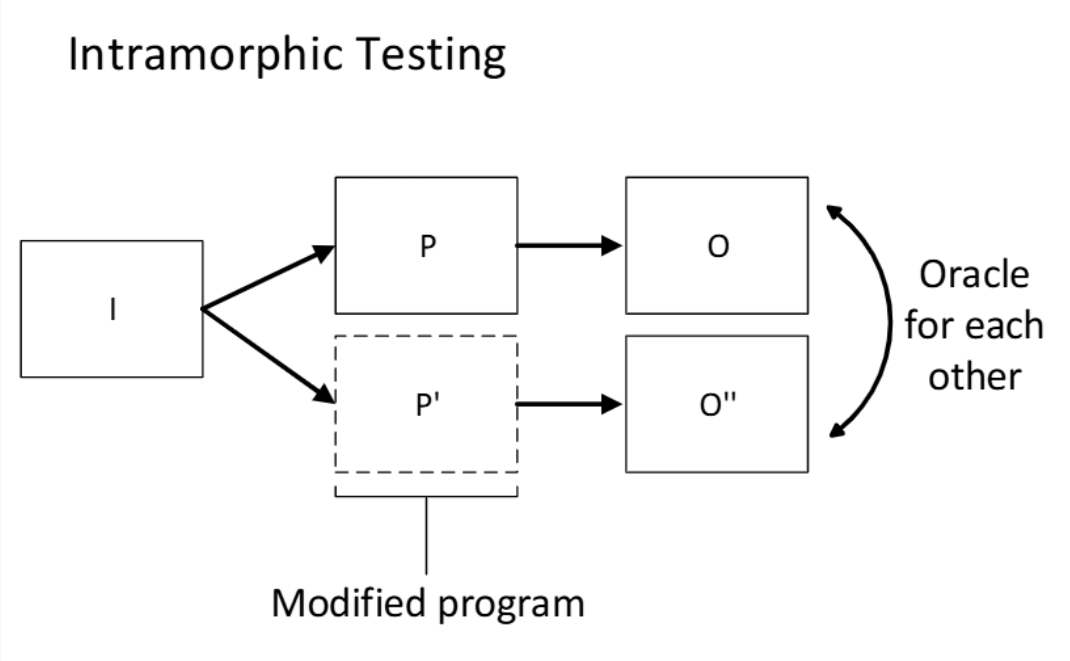
White-box: Intramorphic Testing

Intramorphic Testing



Intramorphic Testing





Intramorphic Testing

- Any inputs can be used for testing
- White-box testing
- Systematic exploration of the program’s internal state

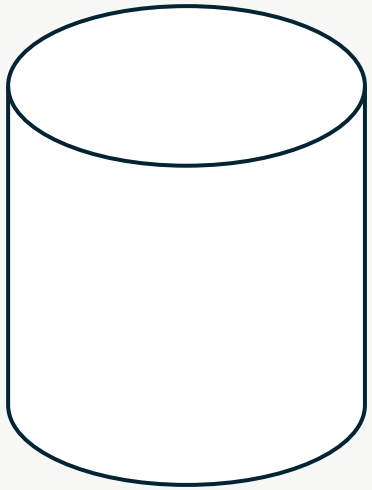
Existing Database Testing Methods

- TLP
- NoREC
- …

```
Q1: SELECT MAX(c0) FROM t0
```

=

```
Q2: SELECT MAX(c0) FROM t0
      WHERE TRUE
```

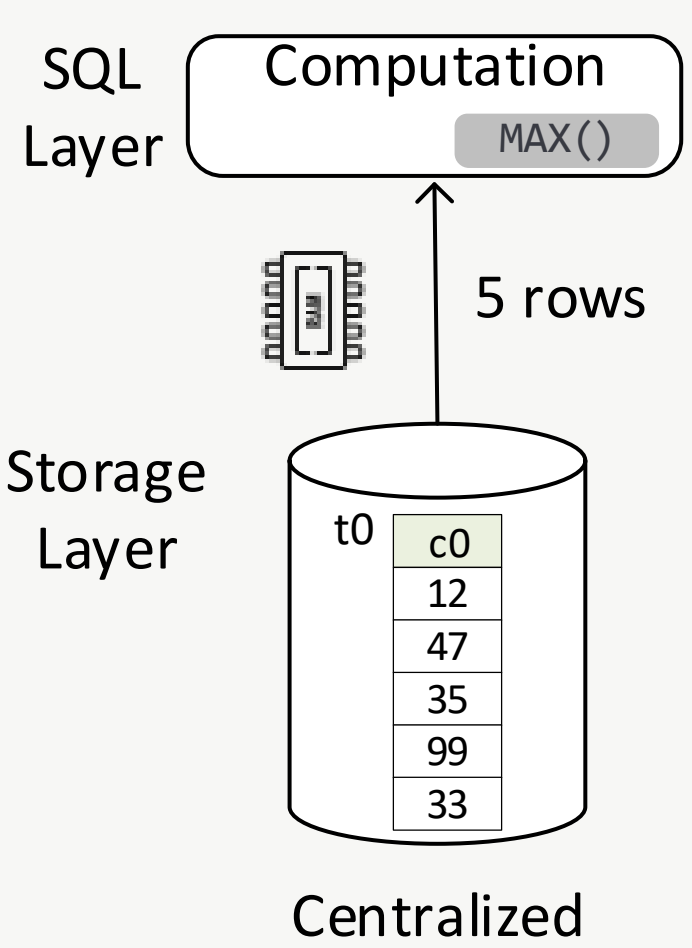


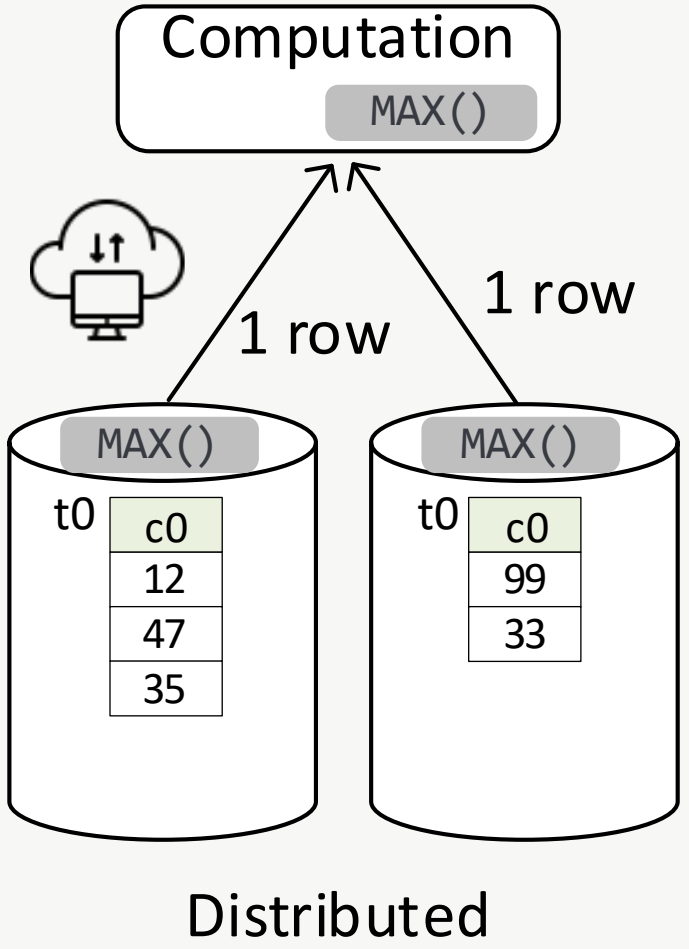
R1: 5

R2: 0

Q1 = Q2, so a bug exists if R1 != R2

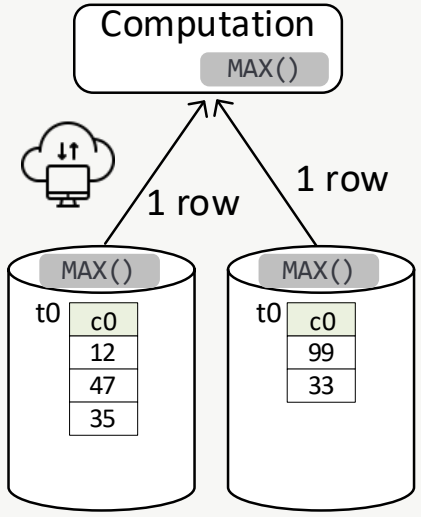
```
SELECT MAX(c0) FROM t0
```





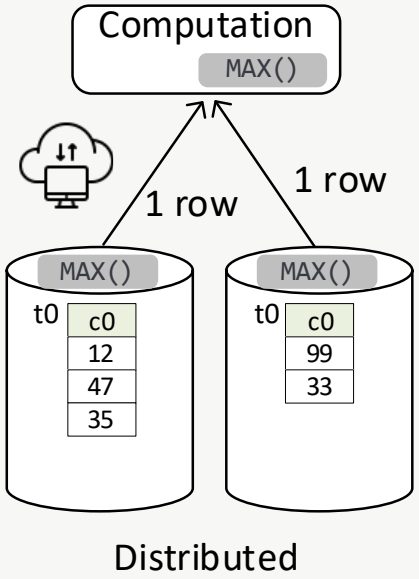
Pushdown Optimization

```
Q1: SELECT MAX(c0) FROM t0
```



Same faulty pushdown path

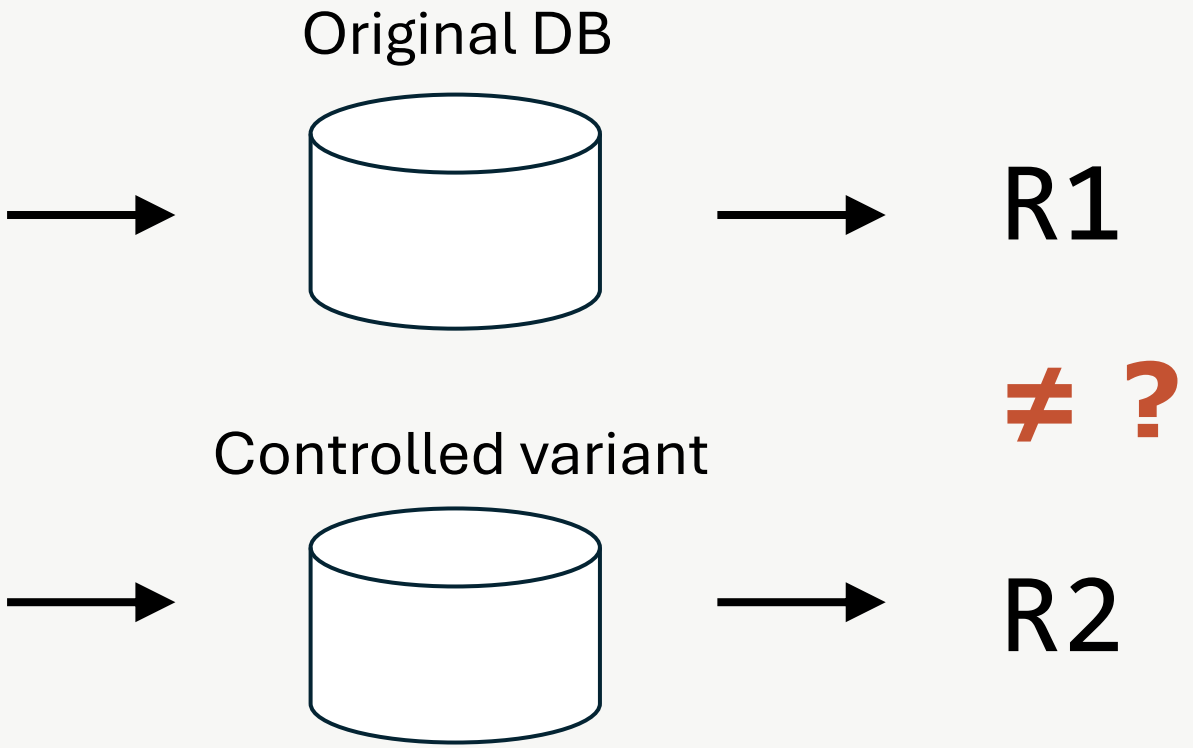
```
Q2: SELECT MAX(c0) FROM t0
      WHERE TRUE
```



Pushdown depends on data storage, network, and other environmental factors.

Our Idea

```
Q1: SELECT MAX(c0)
      FROM t0
```



Construct a DB variant to control pushdown

How to Construct Variants?

DBMS	Aggregate	Distinct	Expression	Function	GroupBy	Join	Topk	Union	Sum
CockroachDB	✓	✓	✓	✓	✓	✓	✓	✓	8
TiDB	✓	✓	✓	✓	✓	✓	✓	✓	8
YugabyteDB	✓	✓	✓	✓	✗	✗	✗	✗	4

```
yb_can_pushdown_max(...) {
   ...
   return true;
}
```

All studied pushdown optimizations are controlled by such scheduling functions

Scheduling Function Identification

Static Matching



push(...)

Functional Validation



Run official test cases

Semantic Checking



Inspect function semantics

YugabyteDB:

62 | 26 | 13

How Accurate It Is

DBMS	All	Pushdown
CockroachDB	12	12
TiDB	26	25
YugabyteDB	13	13
Sum:	51	50

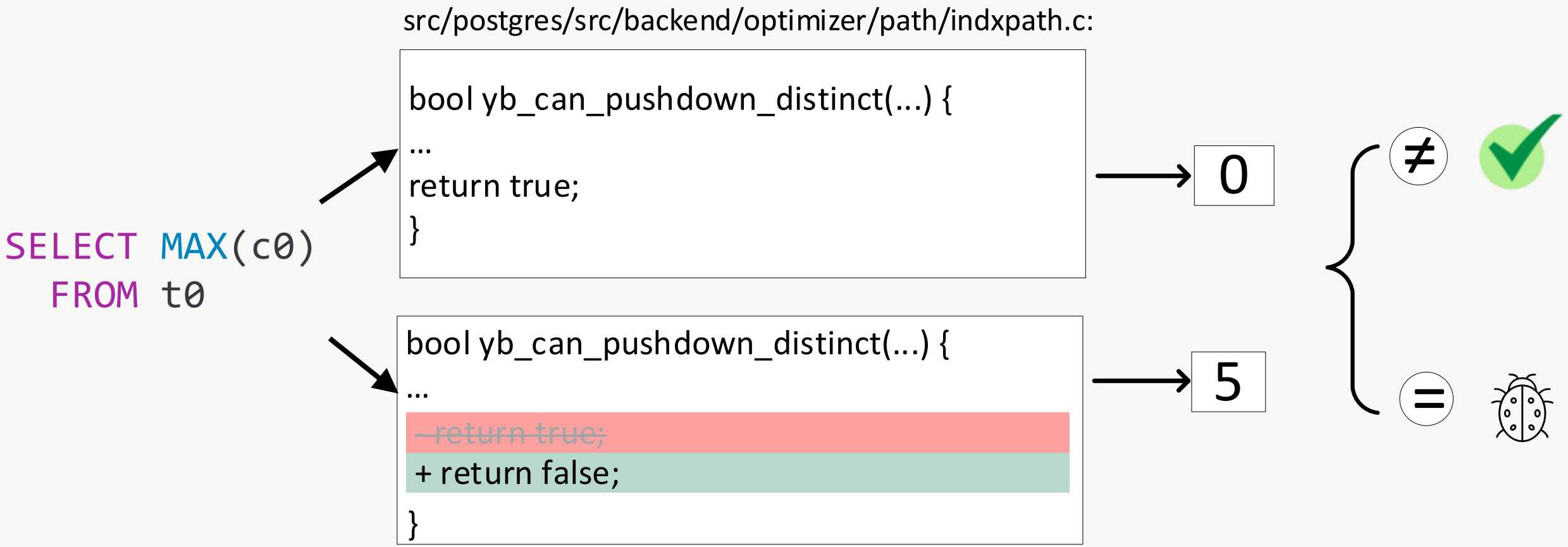
98 % identified functions are scheduling functions

Pushdown

DBMS	Aggregate	Distinct	Expression	Function	GroupBy	Join	Topk	Union	Sum
CockroachDB	✓	✓	✓	✓	✓	✓	✓	✓	8
TiDB	✓	✓	✓	✓	✓	✓	✓	✓	8
YugabyteDB	✓	✓	✓	✓	✗	✗	✗	✗	4

We do not miss any known pushdown optimizations

Testing


src/postgres/src/backend/optimizer/path/indxpath.c:
SELECT MAX(c0)
FROM t0
bool yb_can_pushdown_distinct(...) {
...
return true;
}
bool yb_can_pushdown_distinct(...) {
...
- return true;
+ return false;
}
0
5
≠
=

Evaluation

- 13 logic bugs
- Cover 5 out of the 8 pushdown operators
- 9 bugs Major/High severity

DBMS	Bug ID	Category	Operator	Status
CockroachDB	145108	Logic	Expression	Confirmed
CockroachDB	148279	Logic	Expression	Confirmed
CockroachDB	145888	Crash	-	Fixed
CockroachDB	146065	Crash	-	Fixed
TiDB	60586	Logic	Expression	Fixed
TiDB	60624	Logic	Function	Confirmed
TiDB	60630	Logic	Expression	Confirmed
TiDB	60664	Logic	Function	Confirmed
TiDB	60674	Logic	-	Confirmed
TiDB	60841	Logic	Function	Confirmed
TiDB	60958	Logic	Join	Confirmed
TiDB	60960	Logic	Function	Confirmed
TiDB	61664	Logic	Aggregate	Confirmed
TiDB	61683	Crash	-	Confirmed
TiDB	61684	Crash	-	Fixed
TiDB	61694	Crash	-	Confirmed
TiDB	61715	Crash	-	Fixed
TiDB	61735	Crash	-	Fixed
YugabyteDB	26620	Logic	Aggregate	Confirmed
YugabyteDB	26717	Logic	Distinct	Fixed
YugabyteDB	27031	Logic	Expression	Fixed
YugabyteDB	26753	Crash	-	Confirmed
YugabyteDB	27009	Crash	-	Fixed
YugabyteDB	27029	Crash	-	Confirmed
YugabyteDB	27059	Crash	-	Confirmed
			Sum:	25

Bug Example

```
CREATE TABLE orders(order_id INT, order_date DATE);
CREATE INDEX i0 ON orders(order_date, order_date ASC);
INSERT INTO orders
    VALUES(1, '2025-08-20'), (2, '2025-08-21'), (3, '2025-08-25');
SELECT DISTINCT 'exist' FROM orders WHERE orders.order_date<'2025-09-
01’;   -- {exist} without pushdown, {} with pushdown
```

```
src/postgres/src/backend/optimizer/path/indxpath.c
 -4757,6 +4757,7  yb_can_pushdown_distinct(...) {
--          return true;  // with pushdown
++          return false; // without pushdown
```

General Framework

How to easily apply?

General Framework



Developers



Semantic specifications



Software



One-click run

General Framework

```
...
enum PhaseState {INIT, LOADED, RUNNING}
phase_state;
enum ResourceState {UNAVAILABLE, READY}
resource_state;


if (phase_state = ...
...
switch (resource_state) {
...
```

Specification:

- phase_state: {0,1,2}
- reseource_state: {0,1}
- if (resource_state = 0)
  phase_state <> {1,2}

Available changes to both:

0, 0
1, 1
0, 1
2, 1



香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Questions

Next class: Verification