软件分析与验证前沿

苏亭

软件科学与技术系

Test Oracles

Software Testing (软件测试)

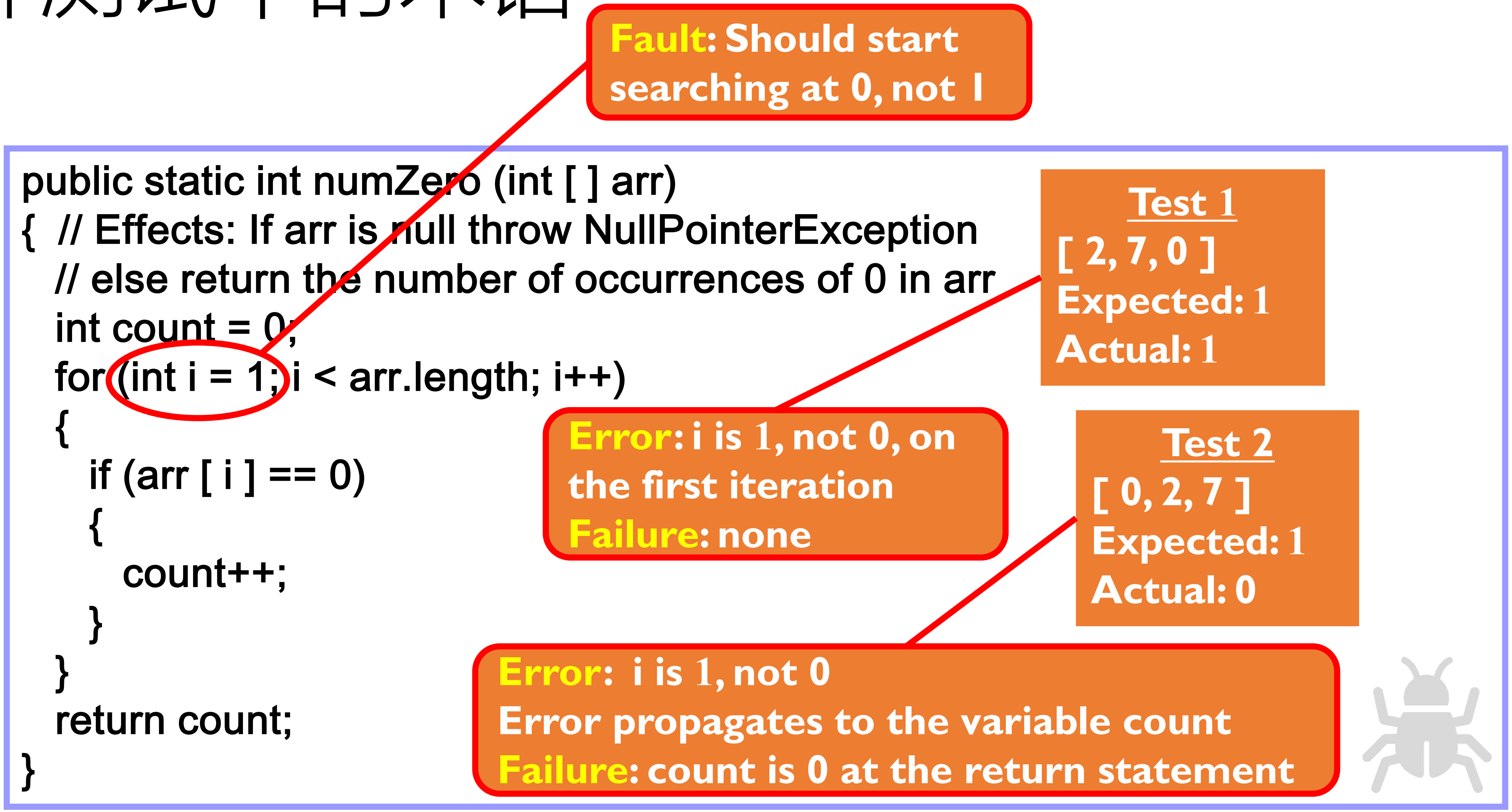
❑ *Assuring* that a system, program or program module is suitable for the purpose for which it was built.

❑ *Examining* the artifacts and the behavior of the software under test by verification and validation.

- ➢ *Are we building the product right? (verification)*
- ➢ *Are we building the right product? (validation)*

❑ **Dijkstra's Law:** Testing can only be used to show the *presence* of errors, but never the *absence* of errors.

软件测试中的术语



```
public static int numZero (int [ ] arr)
{  // Effects: If arr is null throw NullPointerException
   // else return the number of occurrences of 0 in arr
   int count = 0;
   for (int i = 1; i < arr.length; i++)
   {
      if (arr [ i ] == 0)
      {
         count++;
      }
   }
   return count;
}
```

Fault: Should start searching at 0, not 1

Test 1
[2, 7, 0]
Expected: 1
Actual: 1

Error: i is 1, not 0, on the first iteration
Failure: none

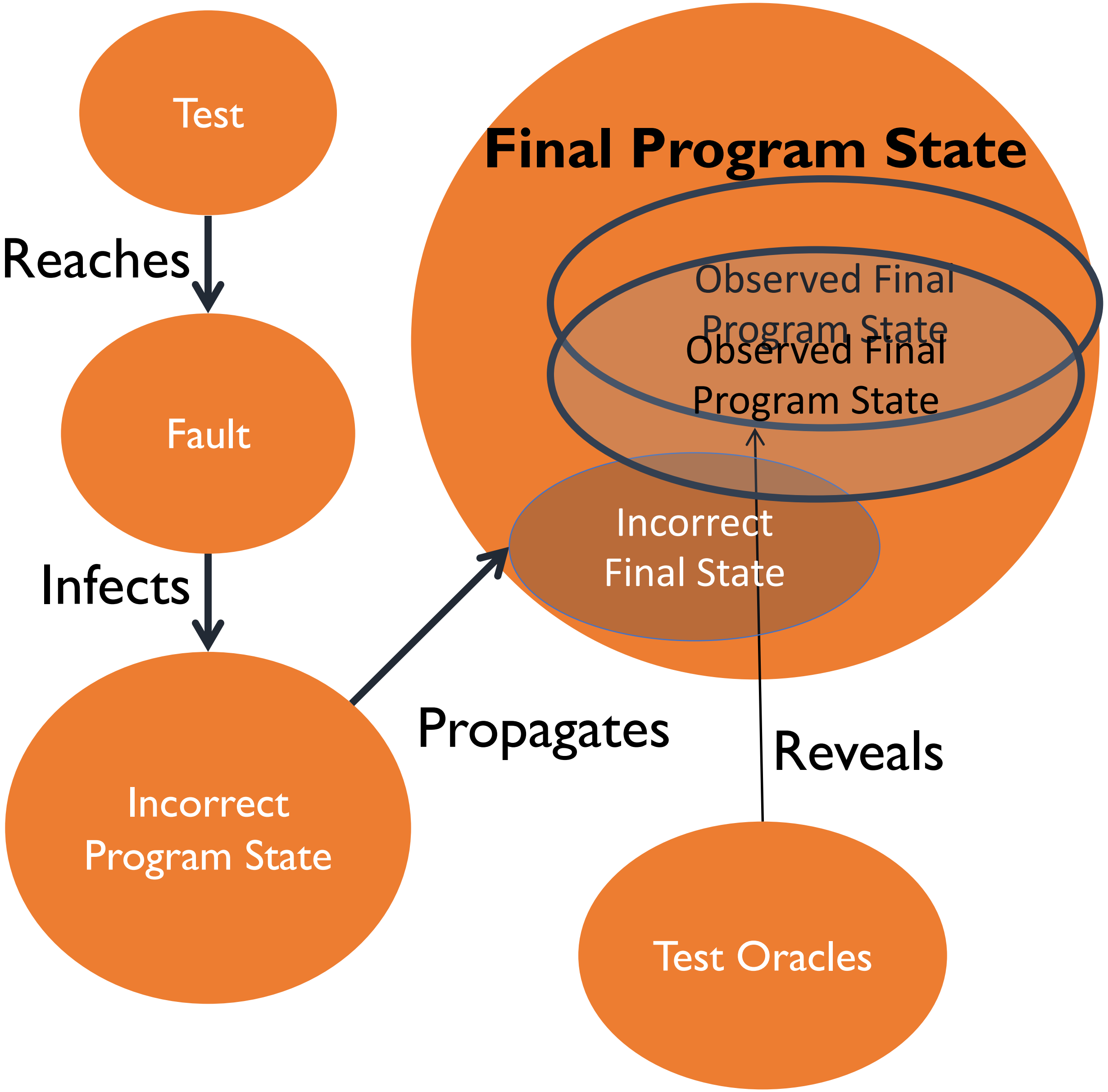
Test 2
[0, 2, 7]
Expected: 1
Actual: 0

Error: i is 1, not 0
Error propagates to the variable count
Failure: count is 0 at the return statement

Fault : A static defect in the software. ***Error*** : An incorrect internal state that is the manifestation of some fault.
Failure : External, incorrect behavior with respect to the requirements of the expected behavior.

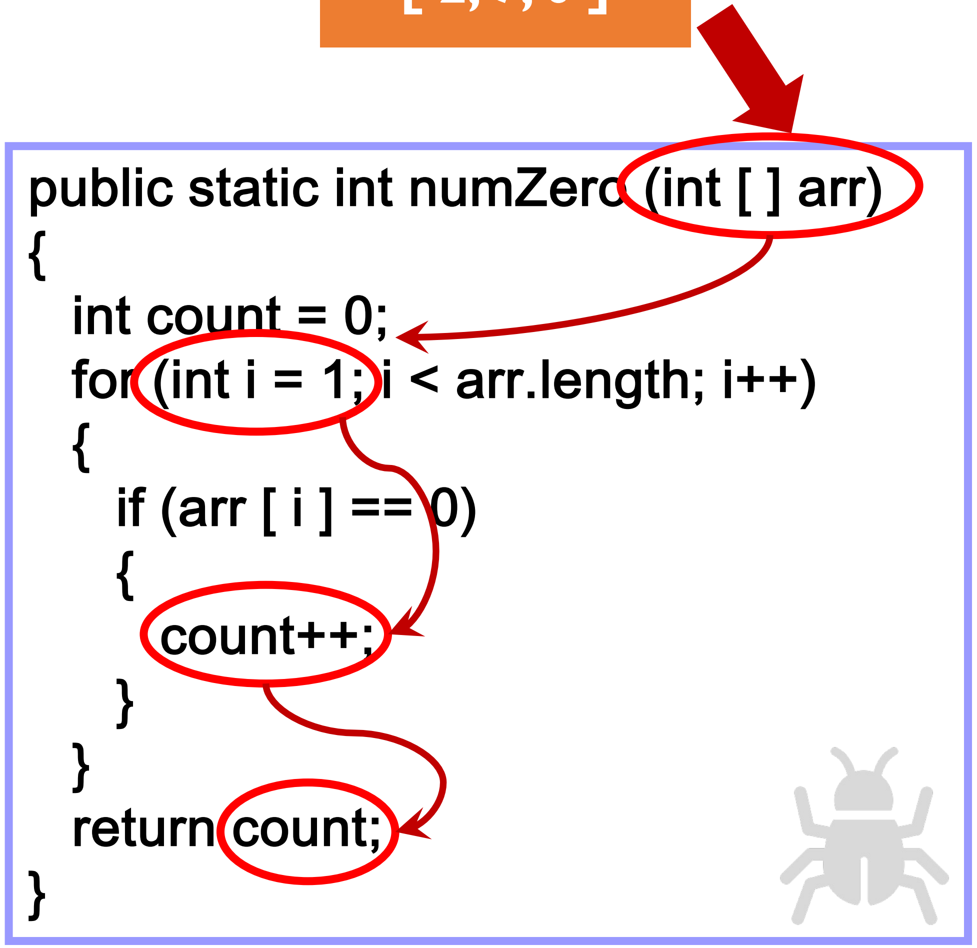
Software Fault and Failure Model (软件错误模型)

- Reachability
- Infection
- Propagation
- Revealability



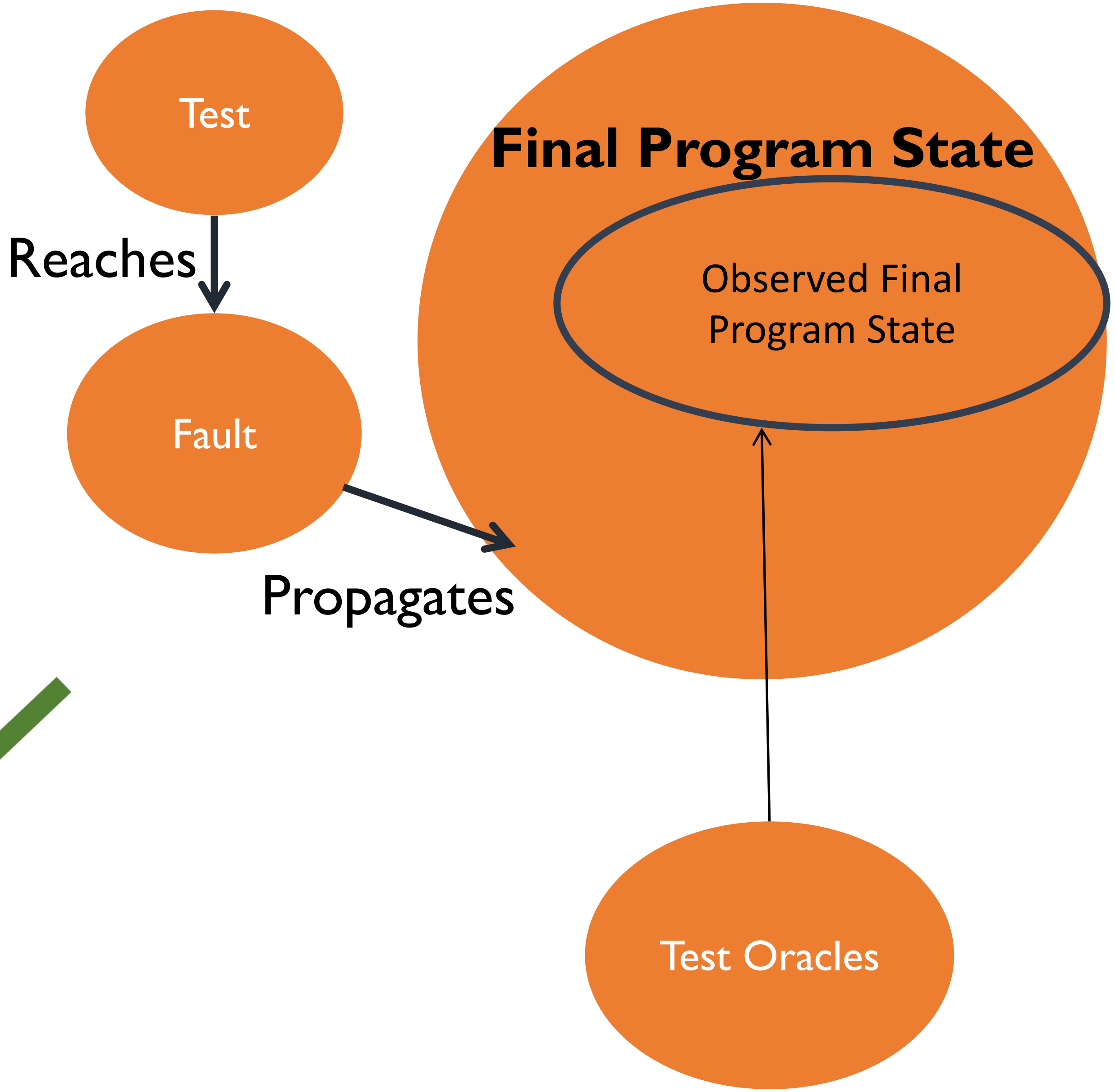
Test 1

[2, 7, 0]



```
public static int numZero (int [ ] arr)
{
   int count = 0;
   for (int i = 1; i < arr.length; i++)
   {
      if (arr [ i ] == 0)
      {
         count++;
      }
   }
   return count;
}
```

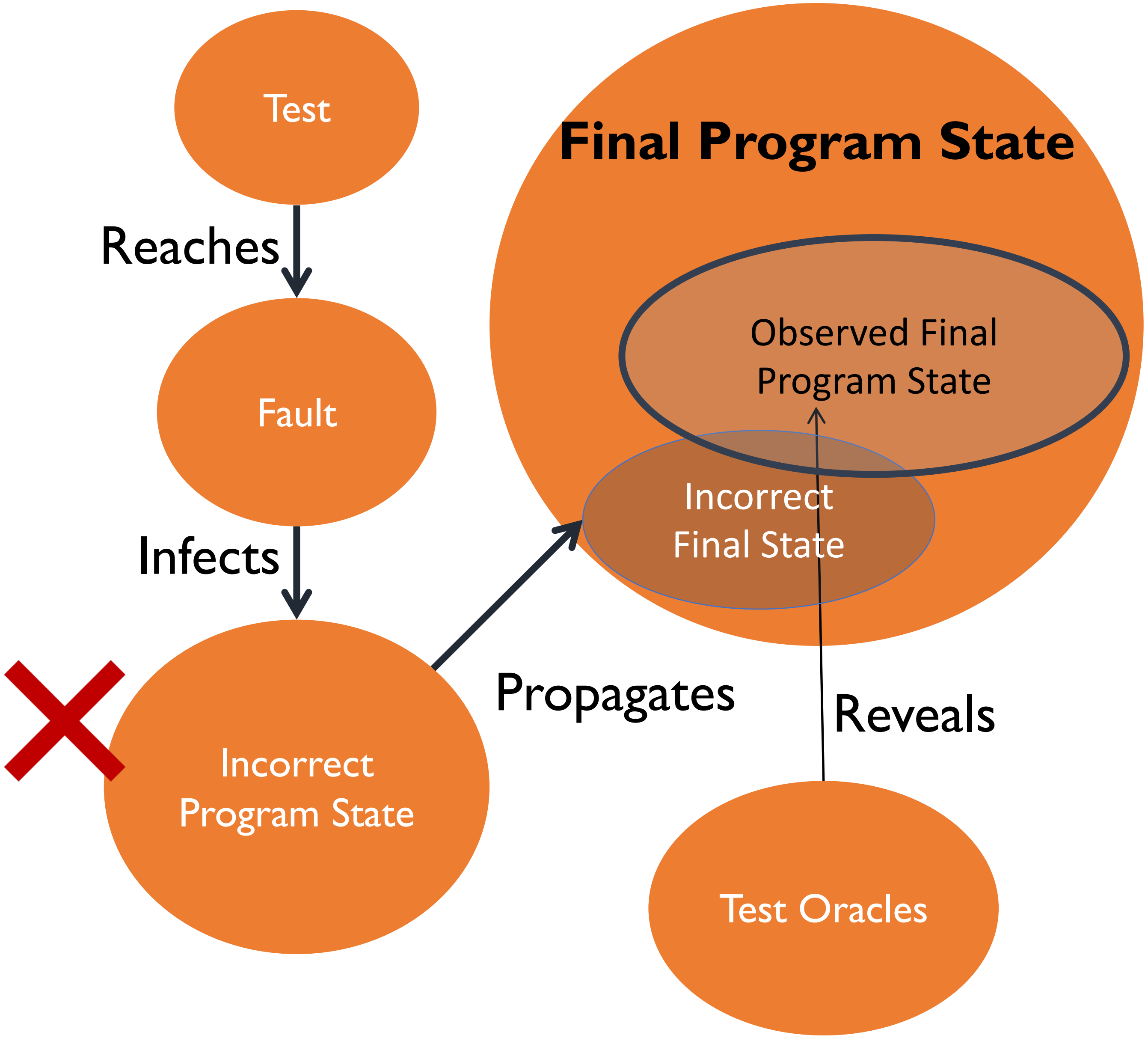
✓



Test 1

[0, 2, 7]

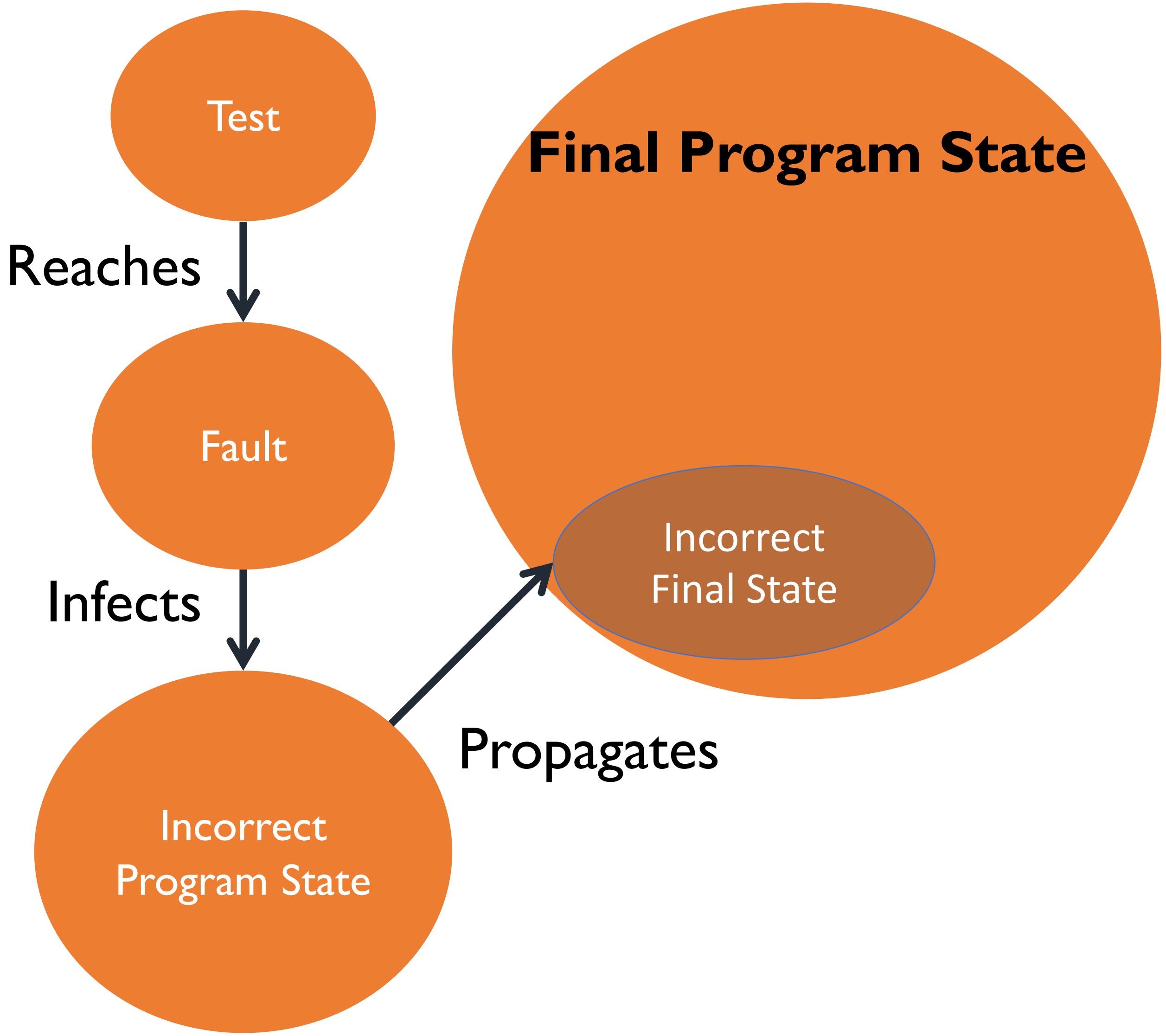
```
public static int numZero (int [ ] arr)
{
   int count = 0;
   for (int i = 1; i < arr.length; i++)
   {
      if (arr [ i ] == 0)
      {
         count++;
      }
   }
   return count;
}
```



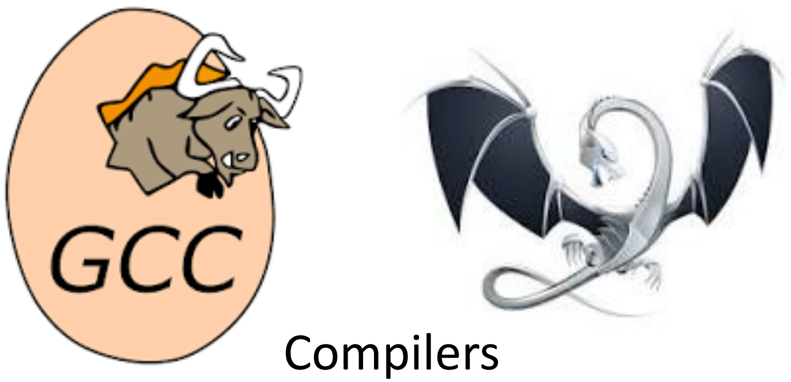
软件测试的挑战在哪里?

软件测试的挑战一:测试数据的设计

- High coverage & diversity
- ~~Redundant~~
- ~~Illegal~~



```
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```



Compilers

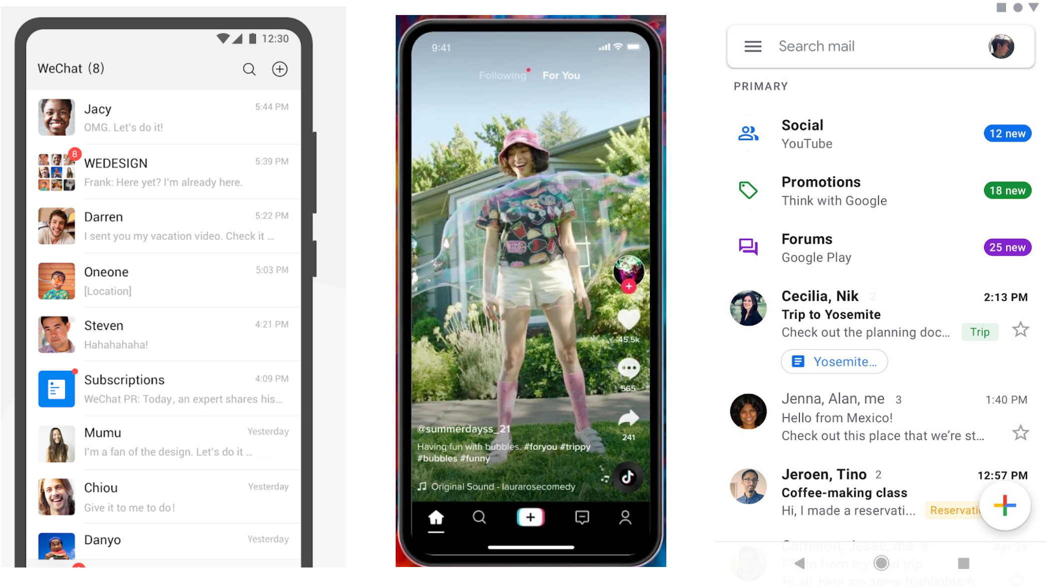
test_input := *event* *

event := *action* (*view* , *text*) | ...

action := click | edit | drag | ...

view := Button | TextField | ...

text := anyString



12:30

WeChat (8)

Jacy 5:44 PM
OMG. Let's do it!

WEDESIGN 5:39 PM
Frank: Here yet? I'm already here.

Darren 5:22 PM
I sent you my vacation video. Check it ...

Oneone 5:03 PM
[Location]

Steven 4:21 PM
Hahahahaha!

Subscriptions 4:09 PM
WeChat PR: Today, an expert shares his...

Mumu Yesterday
I'm a fan of the design. Let's do it ...

Chiou Yesterday
Give it to me to do!

Danyo Yesterday

9:41

Following For You

45.5k
565
241

@summerdayss_21
Having fun with bubbles. #foryou #trippy #bubbles #funny

♫ Original Sound - laurarosecomedy

Search mail

PRIMARY

Social 12 new
YouTube

Promotions 18 new
Think with Google

Forums 25 new
Google Play

Cecilia, Nik 2 2:13 PM
Trip to Yosemite
Check out the planning doc... Trip ☆
Yosemite...

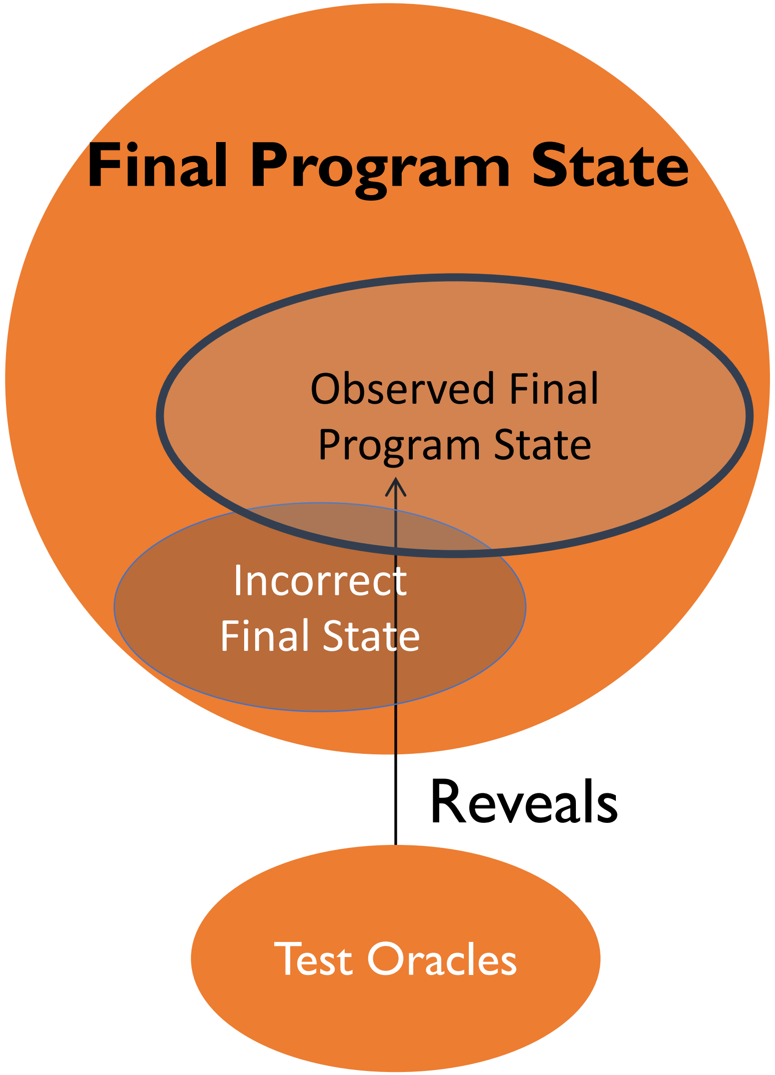
Jenna, Alan, me 3 1:40 PM
Hello from Mexico!
Check out this place that we're st... ☆

Jeroen, Tino 2 12:57 PM
Coffee-making class
Hi, I made a reservati... Reservati

GUI Apps

软件测试的挑战二:测试预言的构造和判定

- How to specify?
- How to check?

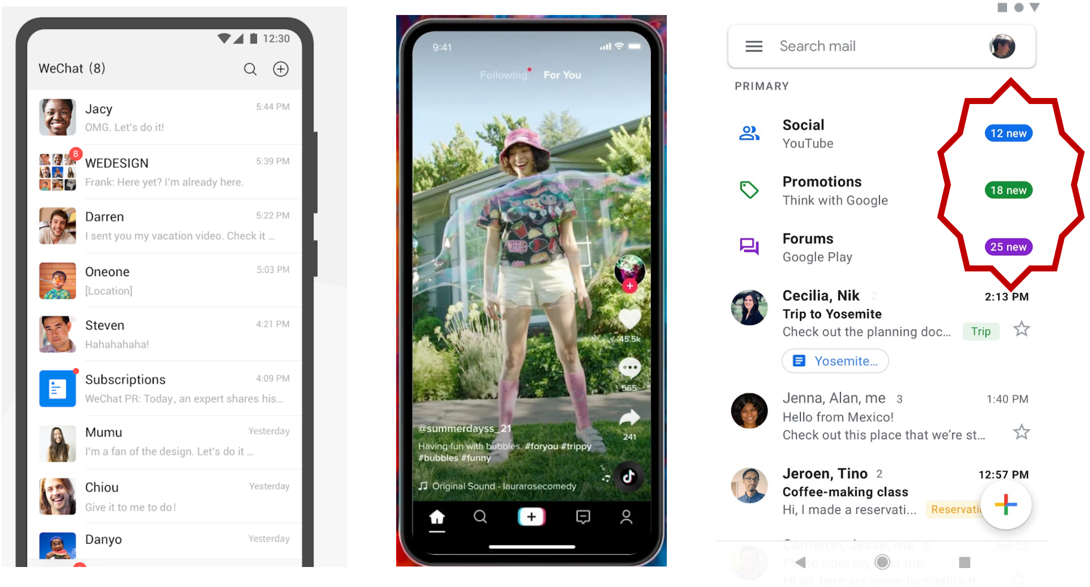


```c
struct tiny { char c; char d; char e; };

void foo(struct tiny x) {
    if (x.c != 1) abort();
    if (x.e != 1) abort();
}

int main() {
    struct tiny s;
    s.c = 1; s.d = 1; s.e = 1;
    foo(s);
    return 0;
}
```

Compilers

$$P' \overset{?}{\equiv} \text{compile}(P)$$

```
test_input  :=  event *
event      :=  action ( view , text )   |  ...
action     :=  click | edit | drag | ...
view       :=  Button | TextField | ...
text       :=  anyString
```



12:30

WeChat (8)

Jacy 5:44 PM
OMG. Let's do it!

WEDESIGN 5:39 PM
Frank: Here yet? I'm already here.

Darren 5:22 PM
I sent you my vacation video. Check it ...

Oneone 5:03 PM
[Location]

Steven 4:21 PM
Hahahahaha!

Subscriptions 4:09 PM
WeChat PR: Today, an expert shares his...

Mumu Yesterday
I'm a fan of the design. Let's do it ...

Chiou Yesterday
Give it to me to do!

Danyo Yesterday

9:41

Following For You

@summerdayss_21
Having fun with bubbles. #foryou #trippy #bubbles #funny

Original Sound - laurarosecomedy

45.5k
565
241

Search mail

PRIMARY

Social 12 new
YouTube

Promotions 18 new
Think with Google

Forums 25 new
Google Play

Cecilia, Nik 2 2:13 PM
Trip to Yosemite
Check out the planning doc... Trip
Yosemite...

Jenna, Alan, me 3 1:40 PM
Hello from Mexico!
Check out this place that we're st...

Jeroen, Tino 2 12:57 PM
Coffee-making class
Hi, I made a reservati... Reservati

GUI Apps

软件测试的两大挑战

测试数据的设计 (Input Generation)

&

测试预言的构造和判定

(Oracle Construction & Checking)

软件测试的两大挑战

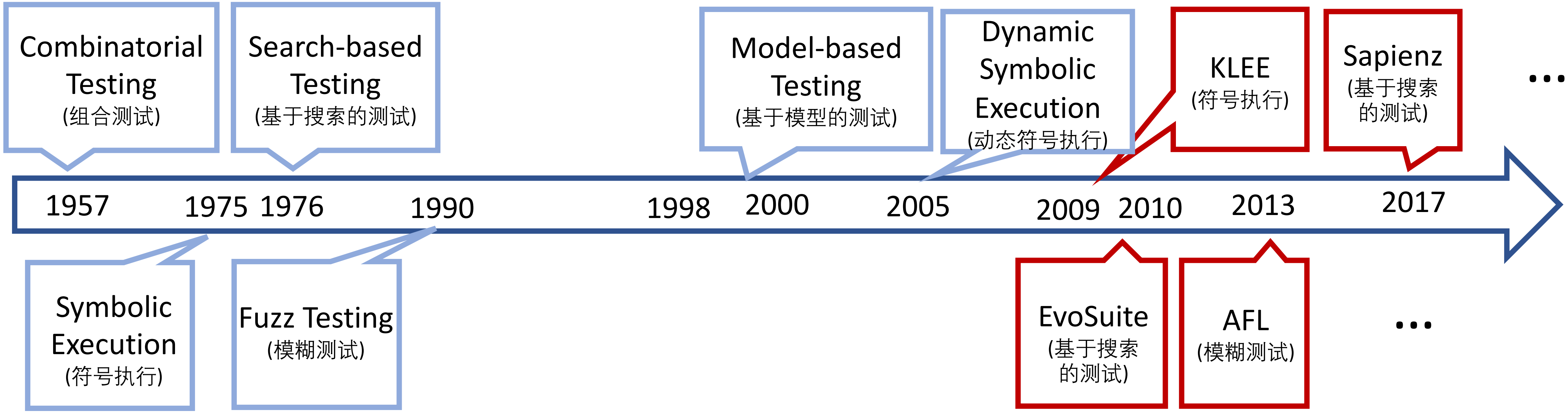
测试数据的设计 (Input Generation)

&

测试预言的构造和判定

(Oracle Construction & Checking)

Techniques for *Input Generation*



Combinatorial Testing (组合测试) 1957

Symbolic Execution (符号执行) 1975

Search-based Testing (基于搜索的测试) 1976

Fuzz Testing (模糊测试) 1990

1998

Model-based Testing (基于模型的测试) 2000

Dynamic Symbolic Execution (动态符号执行) 2005

KLEE (符号执行) 2009

EvoSuite (基于搜索的测试) 2010

AFL (模糊测试) 2013

Sapienz (基于搜索的测试) 2017

...

...

* First proposed, Became popular.

软件测试的两大挑战

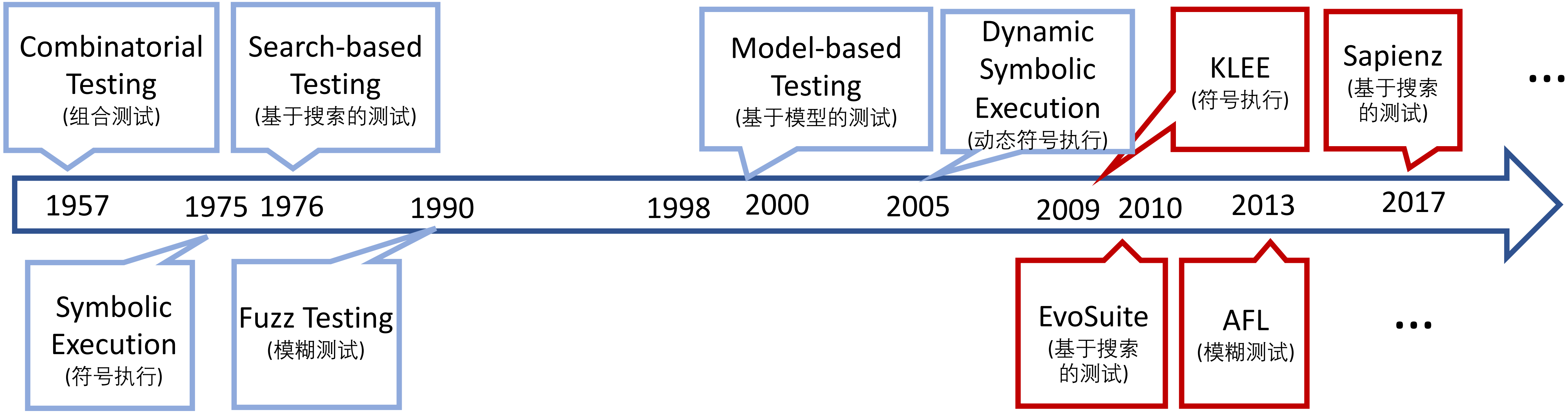
测试数据的设计 (Input Generation)

&

测试预言的构造和判定

(Oracle Construction & Checking)

Techniques for *Oracle Construction and Checking*



Combinatorial Testing (组合测试)

Search-based Testing (基于搜索的测试)

Model-based Testing (基于模型的测试)

Dynamic Symbolic Execution (动态符号执行)

KLEE (符号执行)

Sapienz (基于搜索的测试)

…

1957 1975 1976 1990 1998 2000 2005 2009 2010 2013 2017

Symbolic Execution (符号执行)

Fuzz Testing (模糊测试)

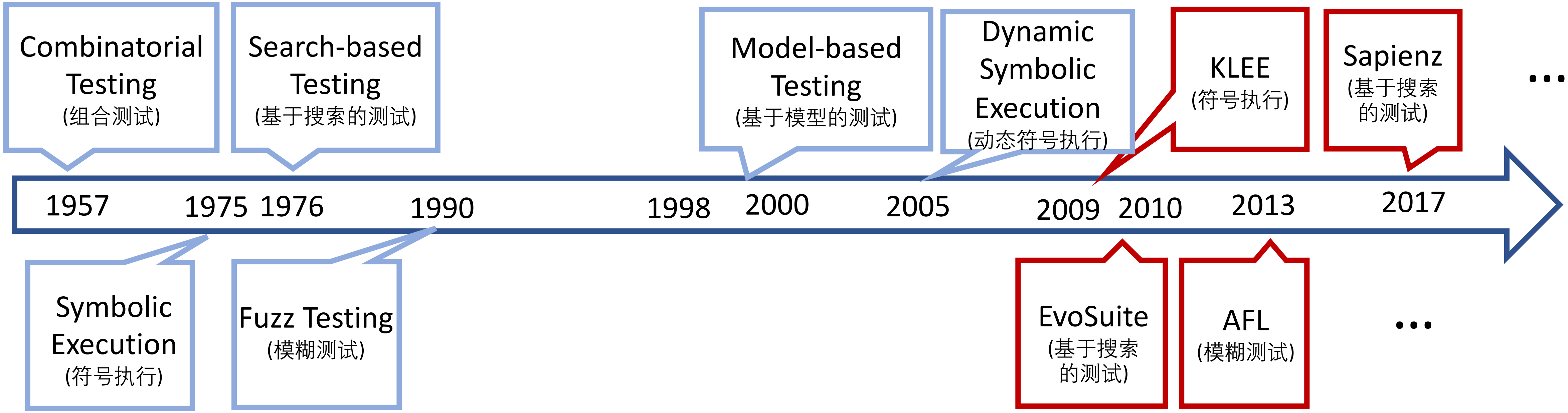
EvoSuite (基于搜索的测试)

AFL (模糊测试)

…

* First proposed, Became popular.

Techniques for *Oracle Construction and Checking*



Combinatorial Testing (组合测试)

Search-based Testing (基于搜索的测试)

Model-based Testing (基于模型的测试)

Dynamic Symbolic Execution (动态符号执行)

KLEE (符号执行)

Sapienz (基于搜索的测试)

…

1957 1975 1976 1990 1998 2000 2005 2009 2010 2013 2017

Symbolic Execution (符号执行)

Fuzz Testing (模糊测试)

EvoSuite (基于搜索的测试)

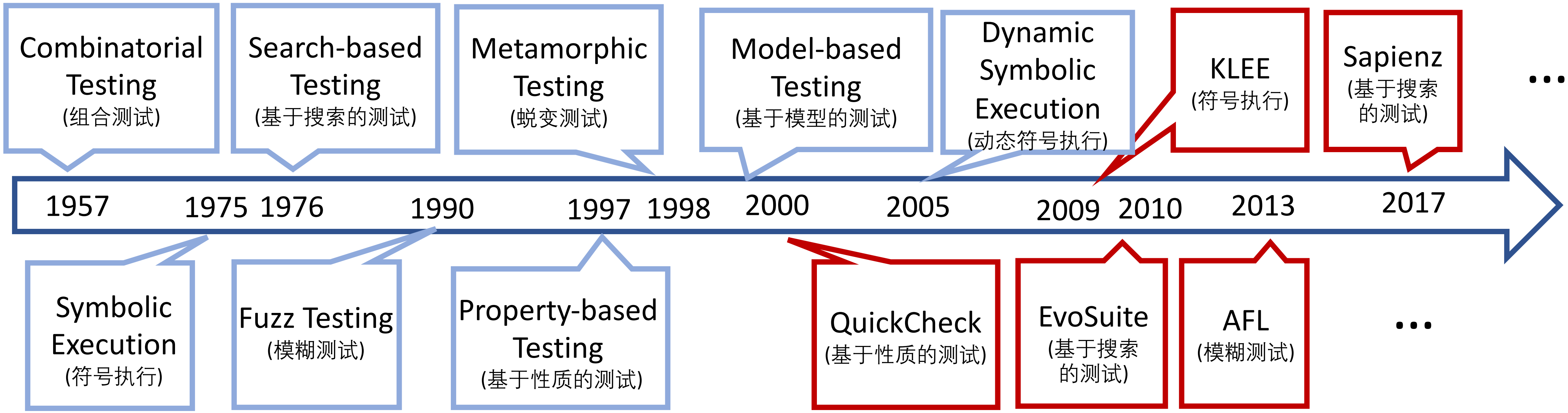
AFL (模糊测试)

…

* First proposed, Became popular.

Crash、 Unhandled Exceptions、 Denial of Service、 Memory leaks、 Buffer Overflows……

Techniques for *Oracle Construction and Checking*



Combinatorial Testing (组合测试)

Search-based Testing (基于搜索的测试)

Metamorphic Testing (蜕变测试)

Model-based Testing (基于模型的测试)

Dynamic Symbolic Execution (动态符号执行)

KLEE (符号执行)

Sapienz (基于搜索的测试)

…

1957 1975 1976 1990 1997 1998 2000 2005 2009 2010 2013 2017

Symbolic Execution (符号执行)

Fuzz Testing (模糊测试)

Property-based Testing (基于性质的测试)

QuickCheck (基于性质的测试)

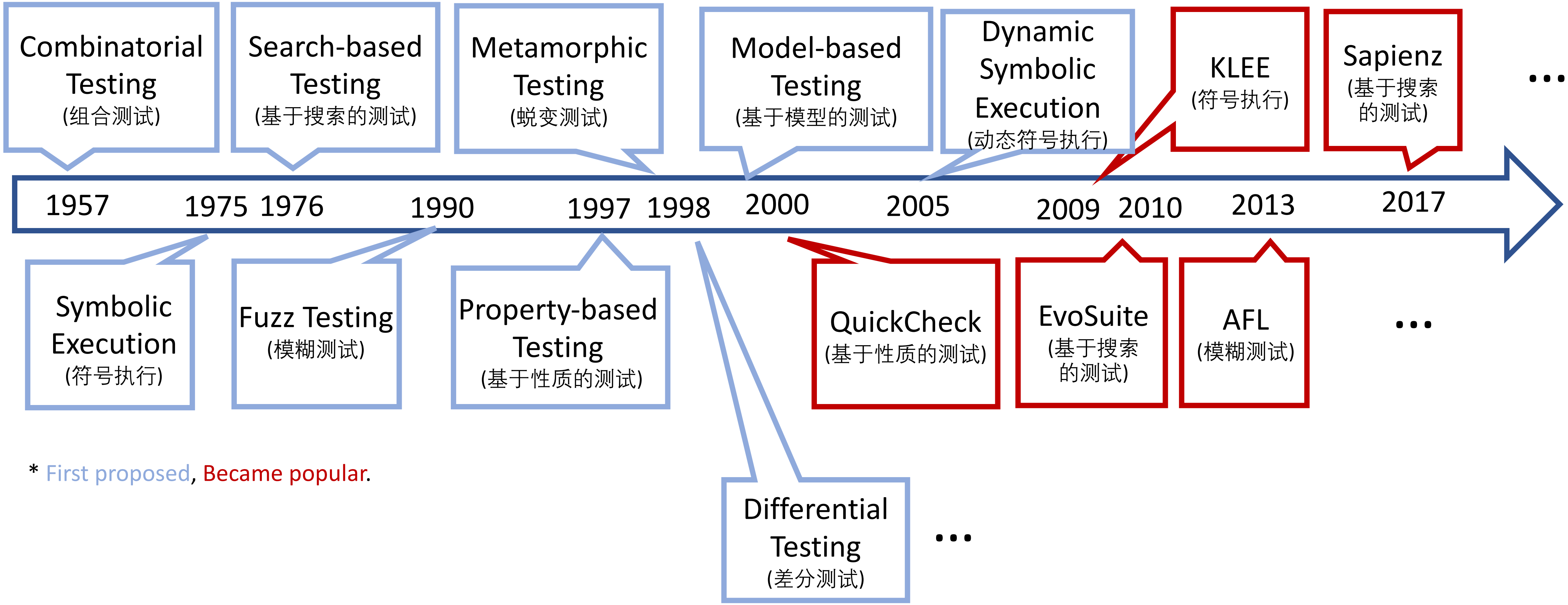
EvoSuite (基于搜索的测试)

AFL (模糊测试)

…

* First proposed, Became popular.

Techniques for *Oracle Construction and Checking*



Combinatorial Testing (组合测试)

Search-based Testing (基于搜索的测试)

Metamorphic Testing (蜕变测试)

Model-based Testing (基于模型的测试)

Dynamic Symbolic Execution (动态符号执行)

KLEE (符号执行)

Sapienz (基于搜索的测试)

…

1957 1975 1976 1990 1997 1998 2000 2005 2009 2010 2013 2017

Symbolic Execution (符号执行)

Fuzz Testing (模糊测试)

Property-based Testing (基于性质的测试)

QuickCheck (基于性质的测试)

EvoSuite (基于搜索的测试)

AFL (模糊测试)

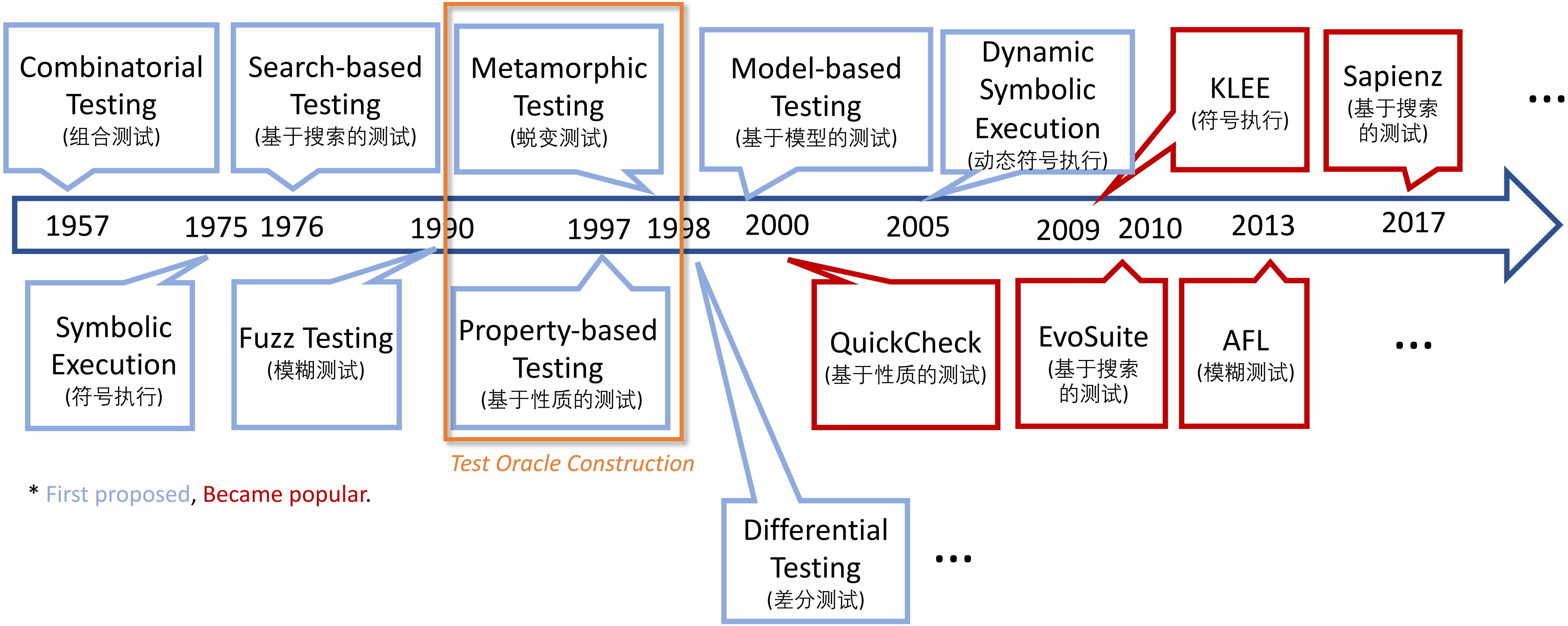
…

Differential Testing (差分测试)

…

* First proposed, Became popular.

Techniques for *Oracle Construction and Checking*



Combinatorial Testing (组合测试)

Search-based Testing (基于搜索的测试)

Metamorphic Testing (蜕变测试)

Model-based Testing (基于模型的测试)

Dynamic Symbolic Execution (动态符号执行)

KLEE (符号执行)

Sapienz (基于搜索的测试)

...

1957 1975 1976 1990 1997 1998 2000 2005 2009 2010 2013 2017

Symbolic Execution (符号执行)

Fuzz Testing (模糊测试)

Property-based Testing (基于性质的测试)

QuickCheck (基于性质的测试)

EvoSuite (基于搜索的测试)

AFL (模糊测试)

...

Test Oracle Construction

Differential Testing (差分测试)

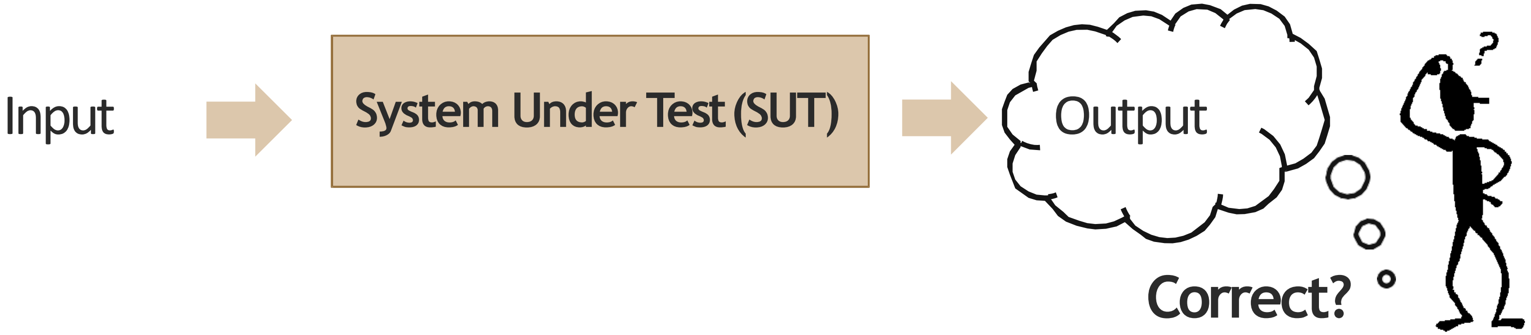
...

* First proposed, Became popular.

软件测试中的Oracle问题

Test Oracle

Test oracle (测试预言)
Mechanism to decide whether a test output is correct or not.

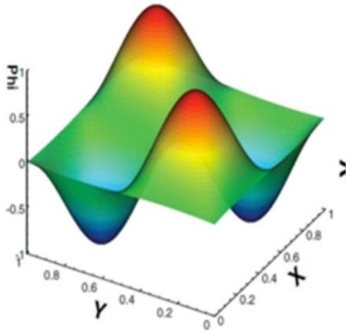


Some slides are borrowed and adapted from “Metamorphic Testing 20 Years Later: A Hands-on Introduction” by Sergio Segura, Zhi Quan Zhou, ICSE 2018, Technical Briefing.

Oracle Problem

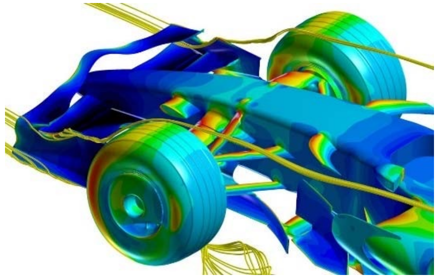
Oracle problem

Sometimes it is not feasible to check the correctness of a test output.



Scientific calculations

Artificial intelligence



Simulation and modelling

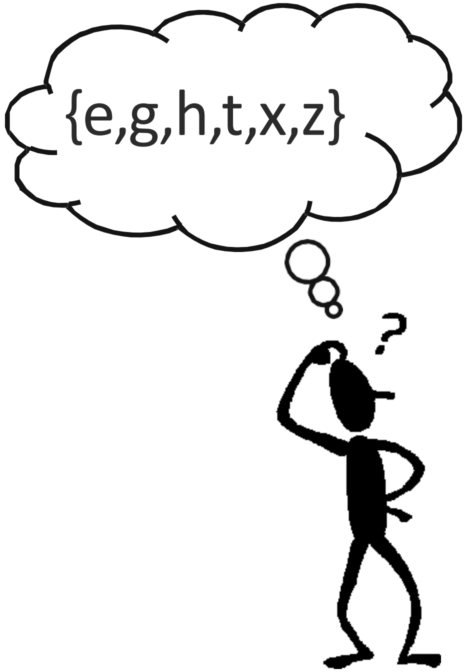
Oracle Problem - Examples

G

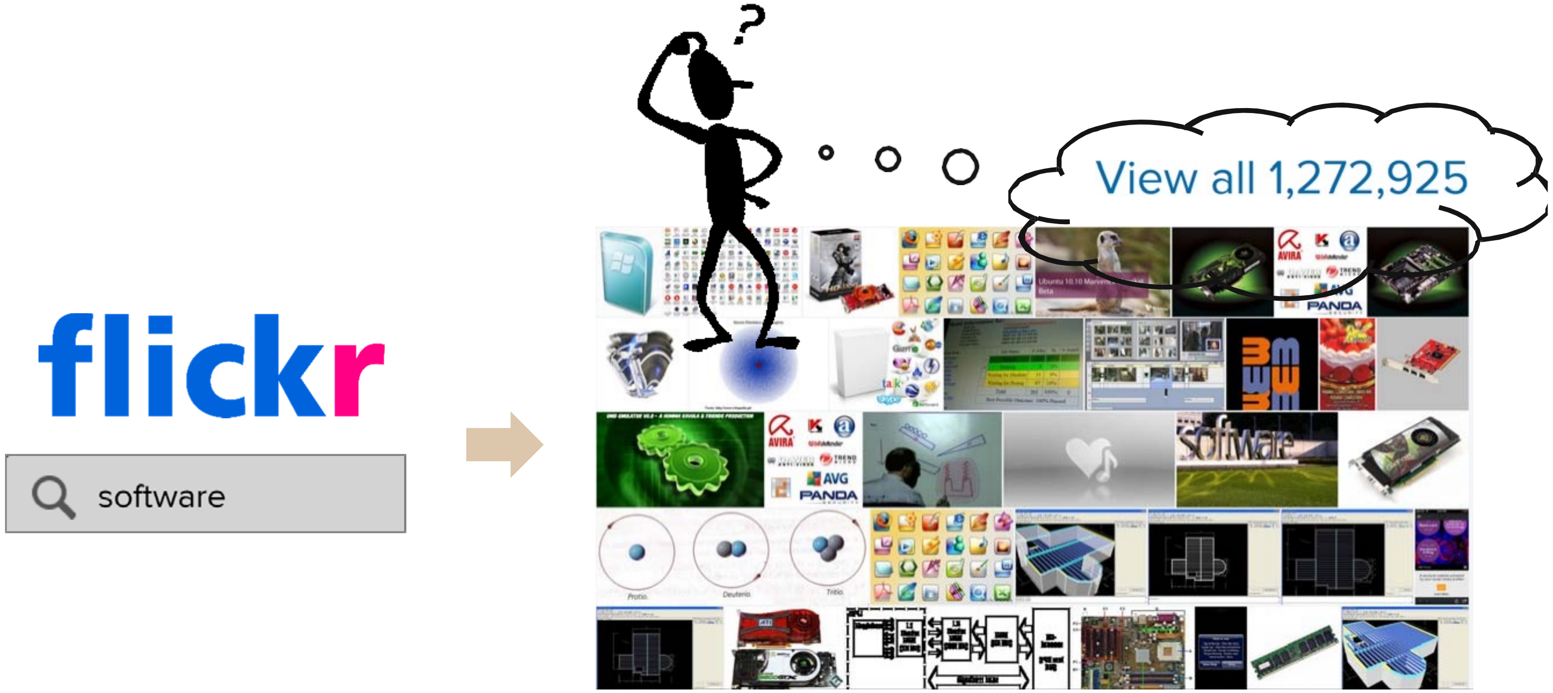
Source= s

Destination=d

shortestPath(G,s,d)



Oracle Problem - Examples



Metamorphic Testing (蜕变测试)

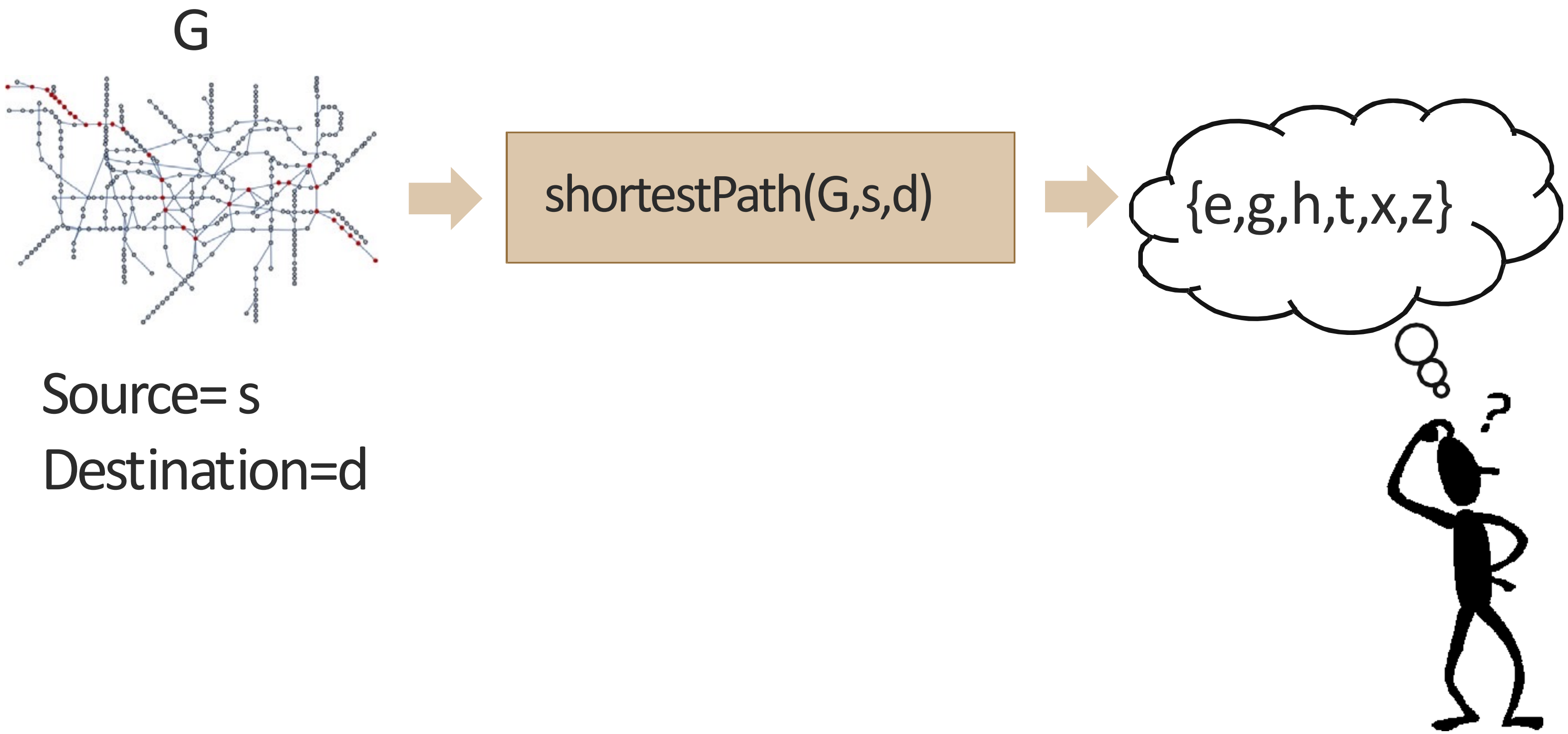
Metamorphic Testing (蜕变测试)

- ❑ An approach to both input generation (测试数据生成) and test result validation (测试结果验证)

- ❑ A central element is a set of metamorphic relations (蜕变关系), which are *necessary* properties (必要性质) of the target function or algorithm *in relation to* multiple inputs and their expected outputs (多个输入和输出之间的关系)

Tsong Yueh Chen, Shing Chi Cheung, and Siu Ming Yiu. 1998. *Metamorphic testing: A new approach for generating next test cases*. Technical Report HKUST-CS98-01. Hong Kong University of Science and Technology, Hong Kong.

Metamorphic Testing - Examples



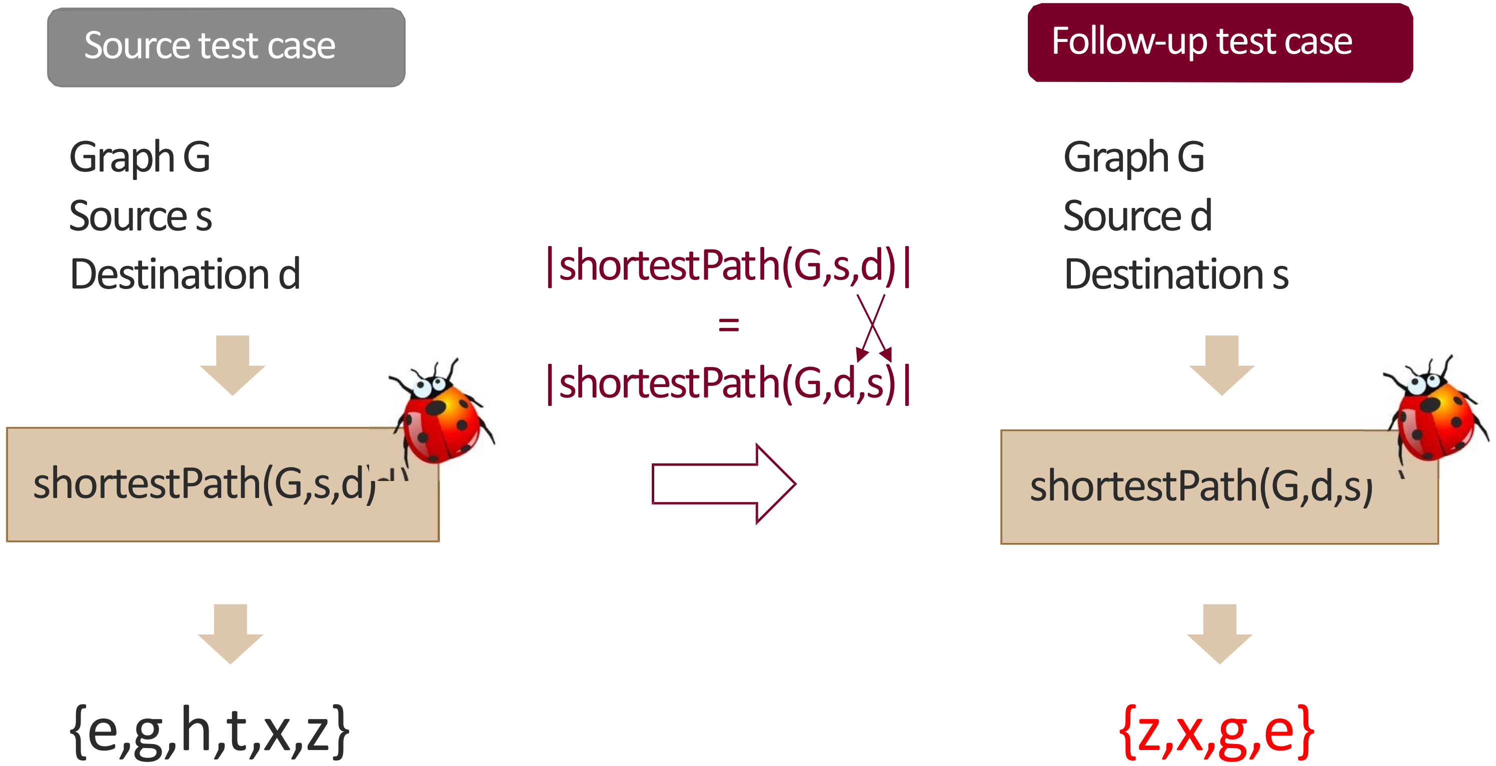
G

shortestPath(G,s,d)

{e,g,h,t,x,z}

Source= s
Destination=d

Metamorphic Testing - Examples



Source test case

Graph G
Source s
Destination d

shortestPath(G,s,d)

{e,g,h,t,x,z}

|shortestPath(G,s,d)|
=
|shortestPath(G,d,s)|

Follow-up test case

Graph G
Source d
Destination s

shortestPath(G,d,s)

{z,x,g,e}

Source test case => seed test case (seed test/seed) ;
Follow-up test case => mutant test case (mutant test/mutant)

Metamorphic Testing - Examples

Source test case

Q_1= “Software”

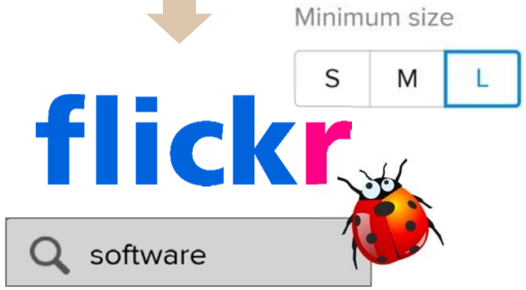
software

View all 1,272,925

If $Q_2 \equiv Q_1$ AND size=large then Count(Q_2) ≤ Count(Q_1)

Follow-up test case

Q_2 = “Software”, size=large



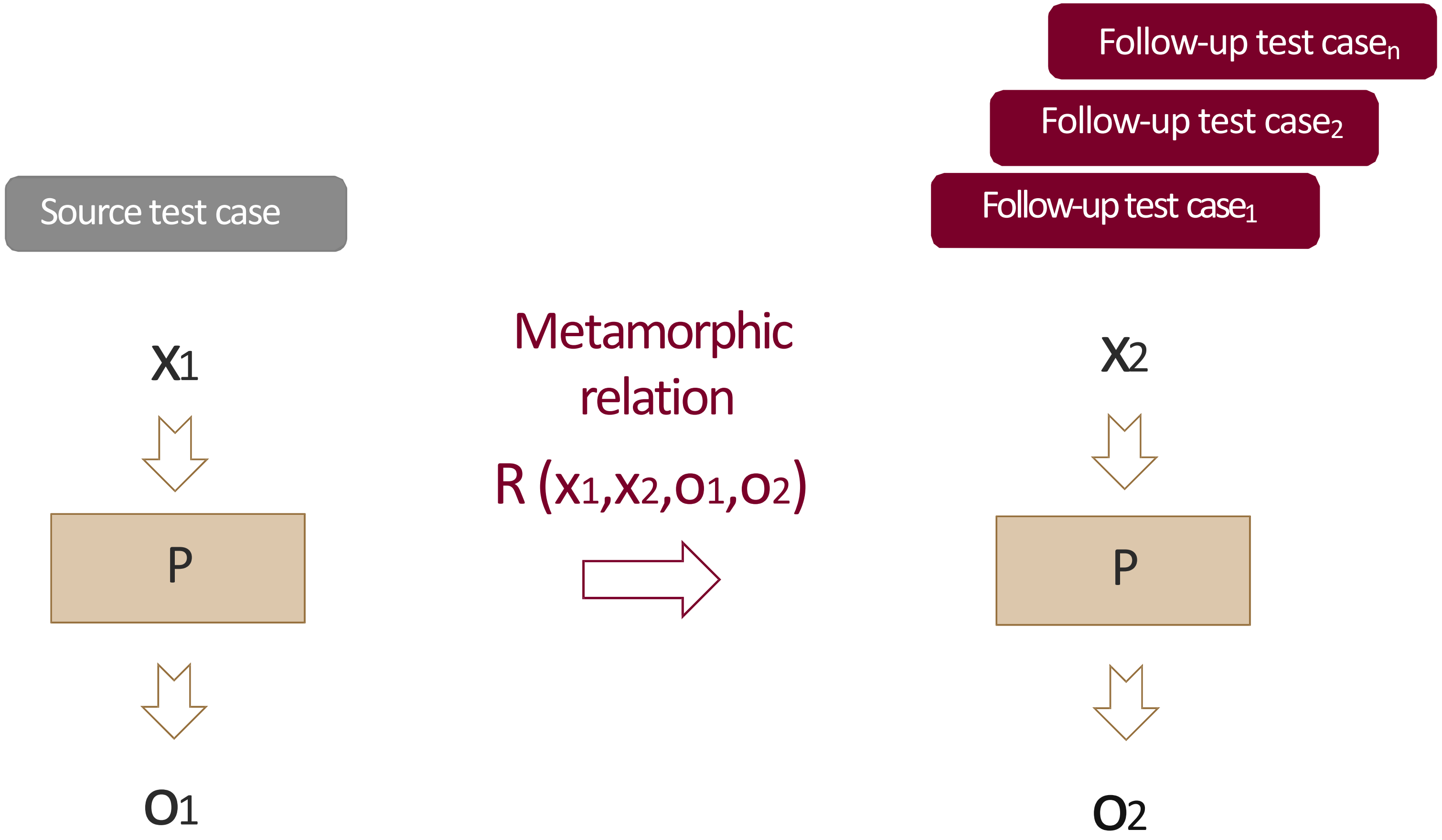
Minimum size

S M [✓ L]

software

View all 1,353,878

Metamorphic Testing - Examples



Source test case

x_1

P

o_1

Metamorphic relation

$R\ (x_1, x_2, o_1, o_2)$

Follow-up test case$_n$

Follow-up test case$_2$

Follow-up test case$_1$

x_2

P

o_2

Metamorphic Testing Process

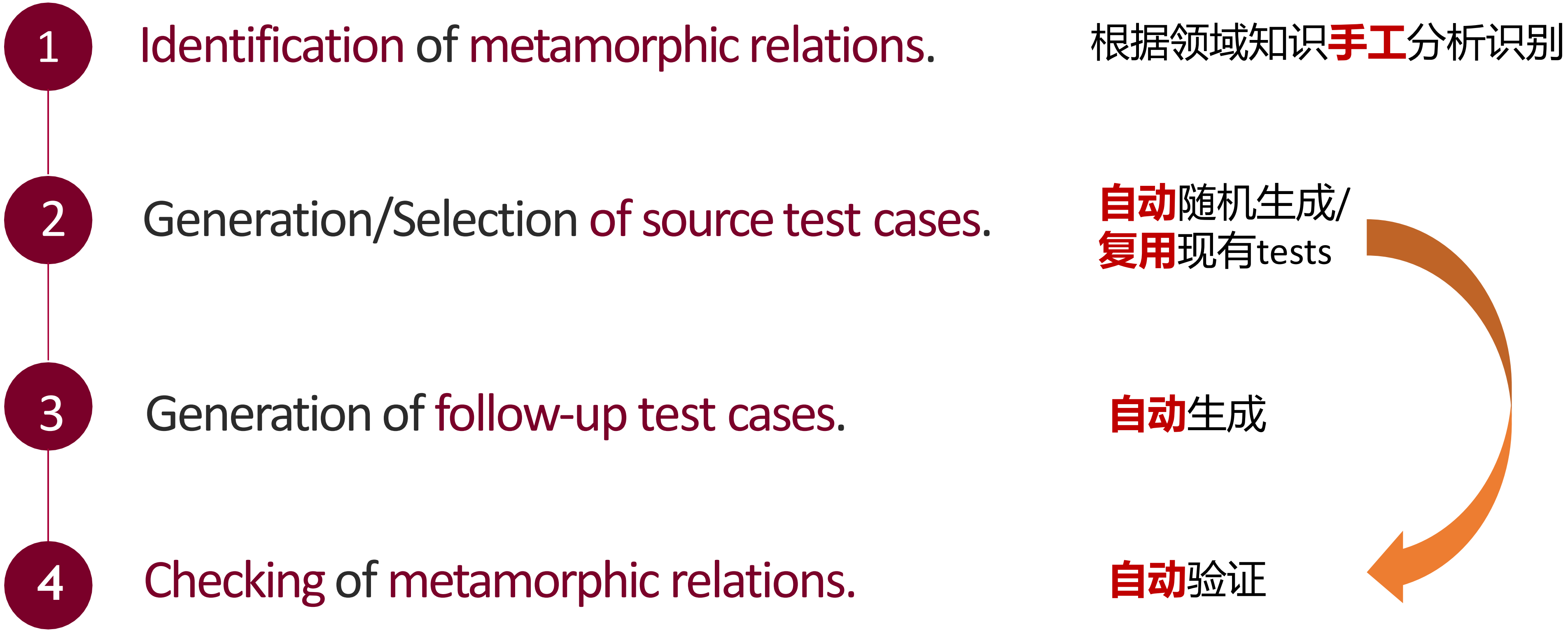
Metamorphic Testing Process

1. Identification of metamorphic relations.
2. Generation/Selection of source test cases.
3. Generation of follow-up test cases.
4. Checking of metamorphic relations.

Metamorphic Testing Process

1. Identification of metamorphic relations. 根据领域知识**手工**分析识别
2. Generation/Selection of source test cases. **自动**随机生成/ **复用**现有tests
3. Generation of follow-up test cases. **自动**生成
4. Checking of metamorphic relations. **自动**验证

Metamorphic Testing Process



1. Identification of metamorphic relations. 根据领域知识**手工**分析识别

2. Generation/Selection of source test cases. **自动**随机生成/**复用**现有tests

3. Generation of follow-up test cases. **自动**生成

4. Checking of metamorphic relations. **自动**验证

Metamorphic Testing Process

1	Identification of metamorphic relations.	根据领域知识**手工**分析识别
2	Generation/Selection of source test cases.	**自动**随机生成/ **复用**现有tests
3	Generation of follow-up test cases.	**自动**生成
4	Checking of metamorphic relations.	**自动**验证

Metamorphic Testing -- State-of-the-Art

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 42, NO. 9, SEPTEMBER 2016 805

A Survey on Metamorphic Testing

Sergio Segura, *Member, IEEE*, Gordon Fraser, *Member, IEEE*, Ana B. Sanchez, and Antonio Ruiz-Cortés

Abstract—A test oracle determines whether a test execution reveals a fault, often by comparing the observed program output to the expected output. This is not always practical, for example when a program's input-output relation is complex and difficult to capture formally. *Metamorphic testing* provides an alternative, where correctness is not determined by checking an individual concrete output, but by applying a transformation to a test input and observing how the program output "morphs" into a different one as a result. Since the introduction of such *metamorphic relations* in 1998, many contributions on metamorphic testing have been made, and the technique has seen successful applications in a variety of domains, ranging from web services to computer graphics. This article provides a comprehensive survey on metamorphic testing: It summarises the research results and application areas, and analyses common practice in empirical studies of metamorphic testing as well as the main open challenges.

Index Terms—Metamorphic testing, oracle problem, survey

1 Introduction

SOFTWARE testing is an essential but costly activity applied during software development to detect faults in programs. Testing consists of executing a program with test inputs, and to detect faults there needs to be some procedure by which testers can decide whether the output of the program is correct or not, a so-called *test oracle* [1]. Often, the test oracle consists of comparing an expected output value with the observed output, but this may not always be feasible. For example, consider programs that produce complex output, like complicated numerical simulations, or code generated by a compiler—predicting the correct output for a given input and then comparing it with the observed output may be non-trivial and error-prone. This problem is referred to as the *oracle problem* and it is recognised as one of the fundamental challenges of software testing [1], [2], [3], [4].

Metamorphic testing [5] is a technique conceived to alleviate the oracle problem. It is based on the idea that often it is simpler to reason about relations between outputs of a program, than it is to fully understand or formalise its input-output behaviour. The prototypical example is that of a program that computes the sine function: What is the exact value of $\sin(12)$? Is an observed output of -0.5365 correct? A mathematical property of the sine function states that $\sin(x) = \sin(\pi - x)$, and we can use this to test whether $\sin(12) = \sin(\pi - 12)$ without knowing the concrete values of either sine calculation. This is an example of a *metamorphic relation*: an input transformation that can be used to generate new test cases from existing test data, and an output relation, that compares the outputs produced by a pair of test cases. Metamorphic testing does not only alleviate the oracle problem, but it can also be highly automated.

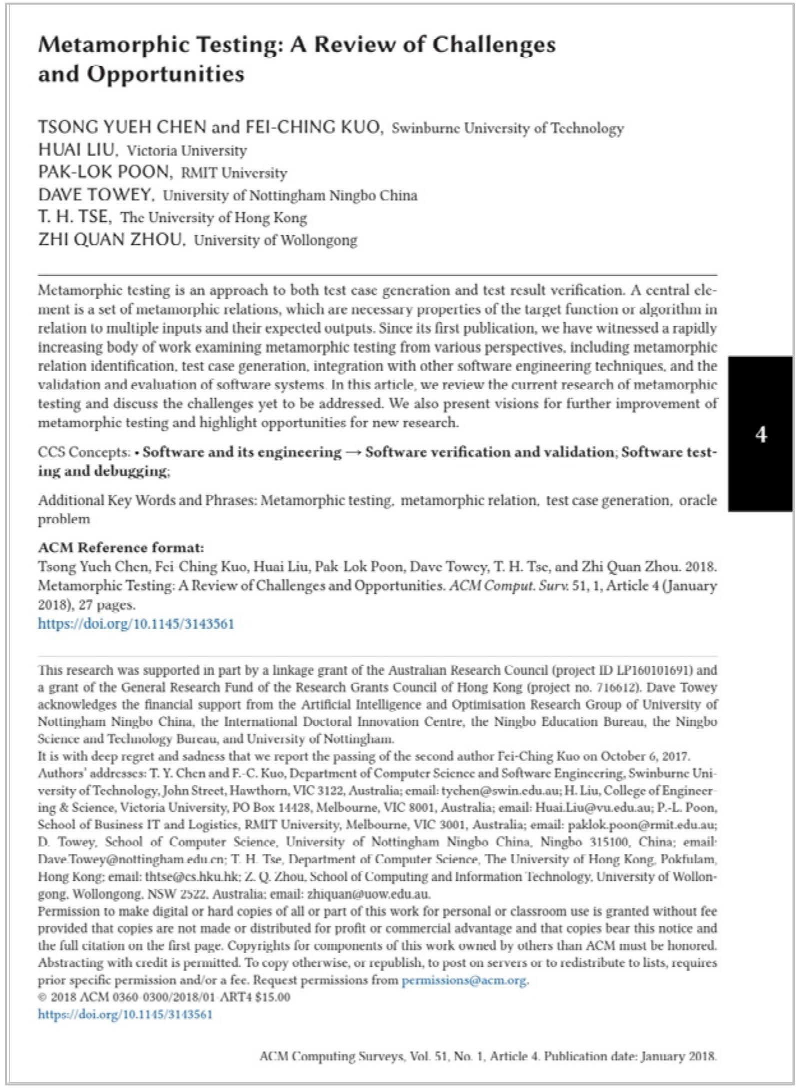
The introduction of metamorphic testing can be traced back to a technical report by Chen et al. [5] published in 1998. However, the use of identity relations to check program outputs can be found in earlier articles on testing of numerical programs [6], [7] and fault tolerance [8]. Since its introduction, the literature on metamorphic testing has flourished with numerous techniques, applications and assessment studies that have not been fully reviewed until now. Although some papers present overviews of metamorphic testing, they are usually the result of the authors' own experience [9], [10], [11], [12], [13], review of selected articles [14], [15], [16] or surveys on related testing topics [3]. At the time of writing this article, the only known survey on metamorphic testing is written in Chinese and was published in 2009[1] [17]. As a result, publications on metamorphic testing remain scattered in the literature, and this hinders the analysis of the state of the art and the identification of new research directions.

In this article, we present an exhaustive survey on metamorphic testing, covering 119 papers published between 1998 and 2015. To provide researchers and practitioners with an entry point, Section 2 contains an introduction to metamorphic testing. All papers were carefully reviewed and classified, and the review methodology followed in our survey as well as a brief summary and analysis of the selected papers are detailed in Section 3. We summarise the state of the art by capturing the main advances on metamorphic testing in Section 4. Across all surveyed papers, we identified more than 12 different application areas, ranging from web services through simulation and modelling to computer graphics (Section 5). Of particular interest for researchers is a detailed analysis of experimental studies and evaluation metrics (Section 6). As a result of our survey, a number of research challenges emerge, providing avenues for future research (Section 7); in particular, there are open questions on how to derive effective metamorphic relations, as well as how to reduce the costs of testing with them.

- *S. Segura, A.B. Sánchez, and A. Ruiz-Cortés are with the Department of Computer Languages and Systems, Universidad de Sevilla, Spain. E-mail: {sergiosegura, anabsanchez, aruiz}@us.es.*
- *G. Fraser is with the Department of Computer Science, University of Sheffield, Sheffield, United Kingdom. E-mail: gordon.fraser@sheffield.ac.uk.*

Manuscript received 11 July 2015; revised 9 Feb. 2016; accepted 14 Feb. 2016. Date of publication 28 Feb. 2016; date of current version 23 Sept. 2016.
Recommended for acceptance by P. Tonella.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSE.2016.2532875

[1] Note that 86 out of the 119 papers reviewed in our survey were published in 2009 or later.



Metamorphic Testing: A Review of Challenges and Opportunities

TSONG YUEH CHEN and FEI-CHING KUO, Swinburne University of Technology
HUAI LIU, Victoria University
PAK-LOK POON, RMIT University
DAVE TOWEY, University of Nottingham Ningbo China
T. H. TSE, The University of Hong Kong
ZHI QUAN ZHOU, University of Wollongong

Metamorphic testing is an approach to both test case generation and test result verification. A central element is a set of metamorphic relations, which are necessary properties of the target function or algorithm in relation to multiple inputs and their expected outputs. Since its first publication, we have witnessed a rapidly increasing body of work examining metamorphic testing from various perspectives, including metamorphic relation identification, test case generation, integration with other software engineering techniques, and the validation and evaluation of software systems. In this article, we review the current research of metamorphic testing and discuss the challenges yet to be addressed. We also present visions for further improvement of metamorphic testing and highlight opportunities for new research.

4

CCS Concepts: • **Software and its engineering → Software verification and validation; Software testing and debugging;**

Additional Key Words and Phrases: Metamorphic testing, metamorphic relation, test case generation, oracle problem

ACM Reference format:
Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, and Zhi Quan Zhou. 2018. Metamorphic Testing: A Review of Challenges and Opportunities. *ACM Comput. Surv.* 51, 1, Article 4 (January 2018), 27 pages.
https://doi.org/10.1145/3143561

This research was supported in part by a linkage grant of the Australian Research Council (project ID LP160101691) and a grant of the General Research Fund of the Research Grants Council of Hong Kong (project no. 716612). Dave Towey acknowledges the financial support from the Artificial Intelligence and Optimisation Research Group of University of Nottingham Ningbo China, the International Doctoral Innovation Centre, the Ningbo Education Bureau, the Ningbo Science and Technology Bureau, and University of Nottingham.

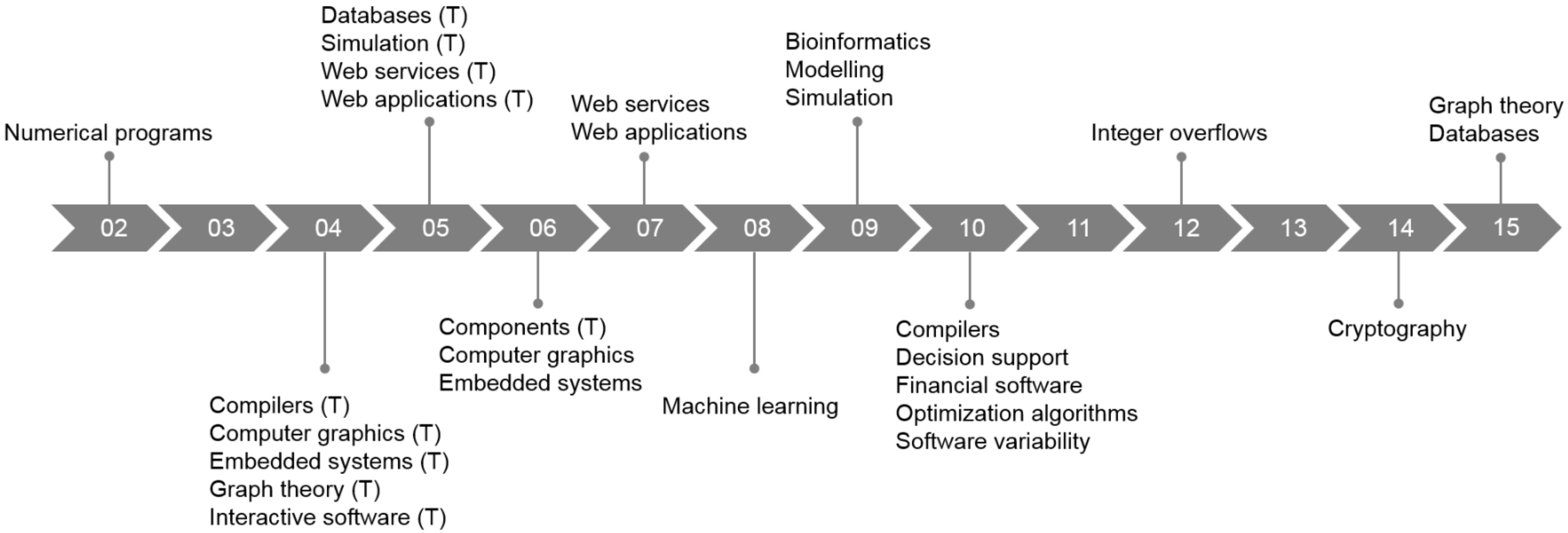
It is with deep regret and sadness that we report the passing of the second author Fei-Ching Kuo on October 6, 2017.

Authors' addresses: T. Y. Chen and F.-C. Kuo, Department of Computer Science and Software Engineering, Swinburne University of Technology, John Street, Hawthorn, VIC 3122, Australia; email: tychen@swin.edu.au; H. Liu, College of Engineering & Science, Victoria University, PO Box 14428, Melbourne, VIC 8001, Australia; email: Huai.Liu@vu.edu.au; P.-L. Poon, School of Business IT and Logistics, RMIT University, Melbourne, VIC 3001, Australia; email: paklok.poon@rmit.edu.au; D. Towey, School of Computer Science, University of Nottingham Ningbo China, Ningbo 315100, China; email: Dave.Towey@nottingham.edu.cn; T. H. Tse, Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong; email: thtse@cs.hku.hk; Z. Q. Zhou, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia; email: zhiquan@uow.edu.au.

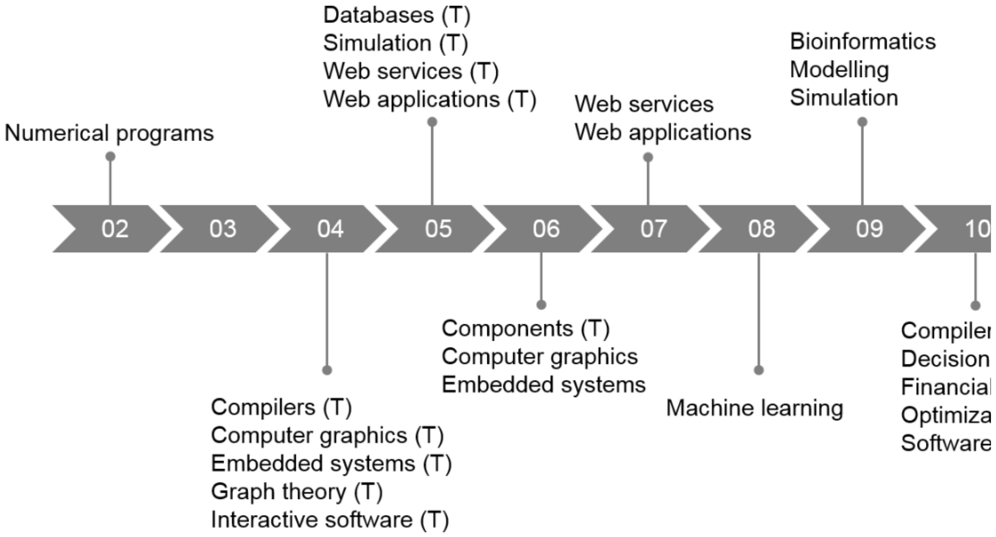
https://doi.org/10.1145/3143561

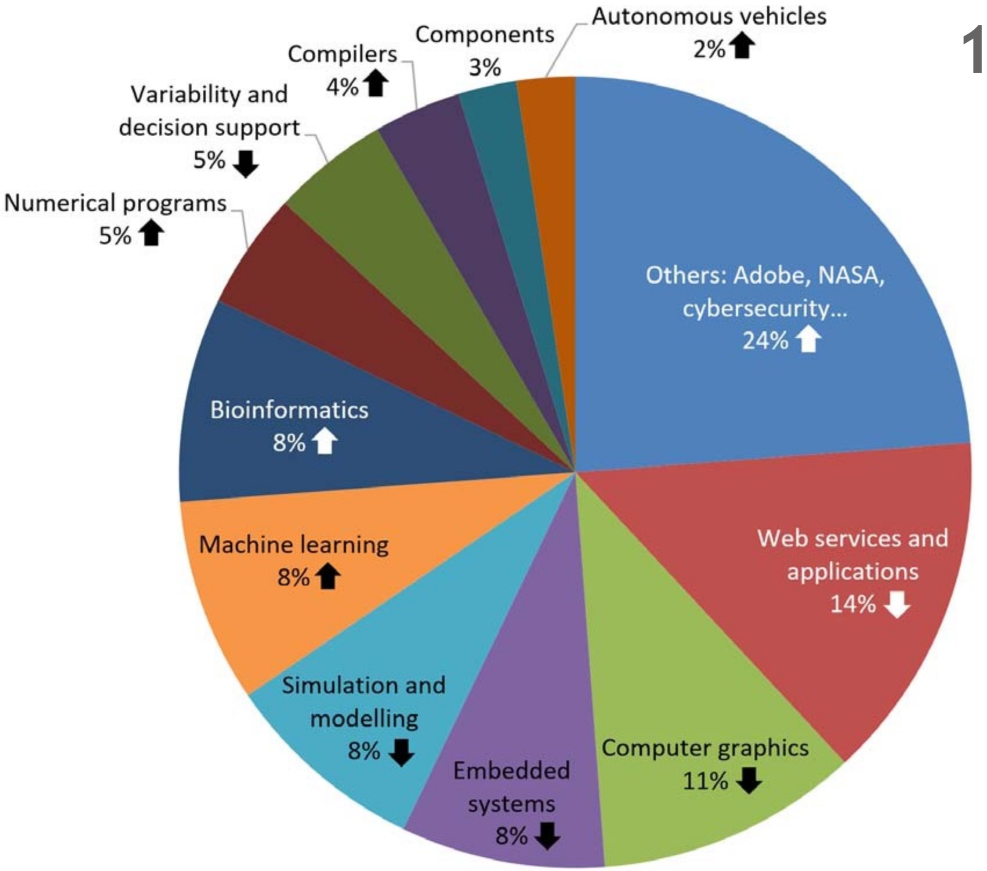
ACM Computing Surveys, Vol. 51, No. 1, Article 4. Publication date: January 2018.

Metamorphic Testing – State-of-the-Art



Metamorphic Testing – State-of-the-Art





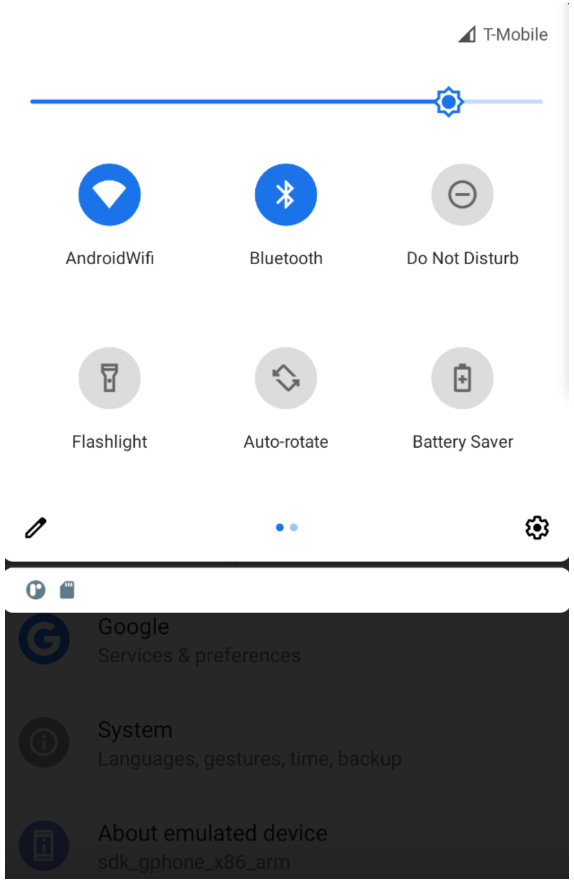
1998 - 2018

84 case studies

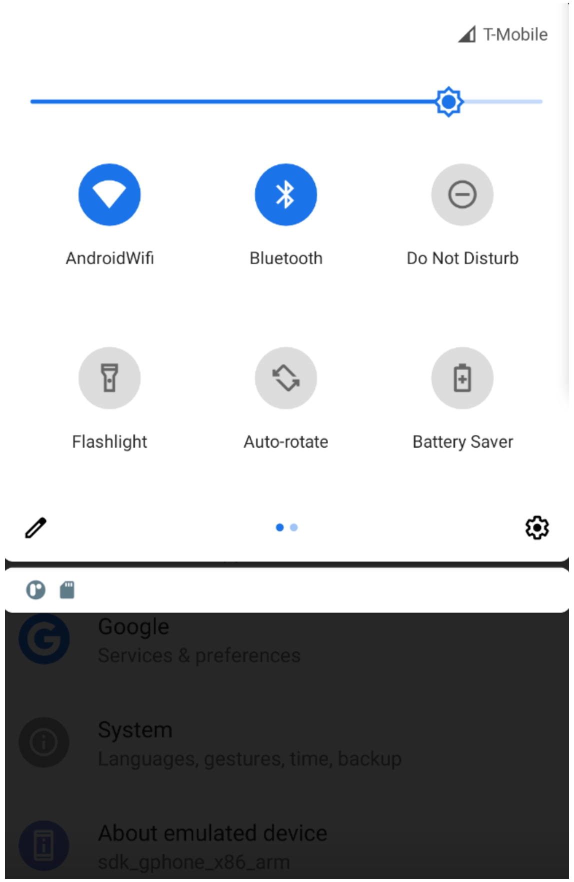
2018-2023年：物体识别系统、自然语言处理模型、数据库管理软件、编译器、自动驾驶模块、移动Apps、领域特定软件（如Datalog分析引擎、SMT求解器）、硬件处理器等。

蜕变测试的探索案例

System Settings in Android



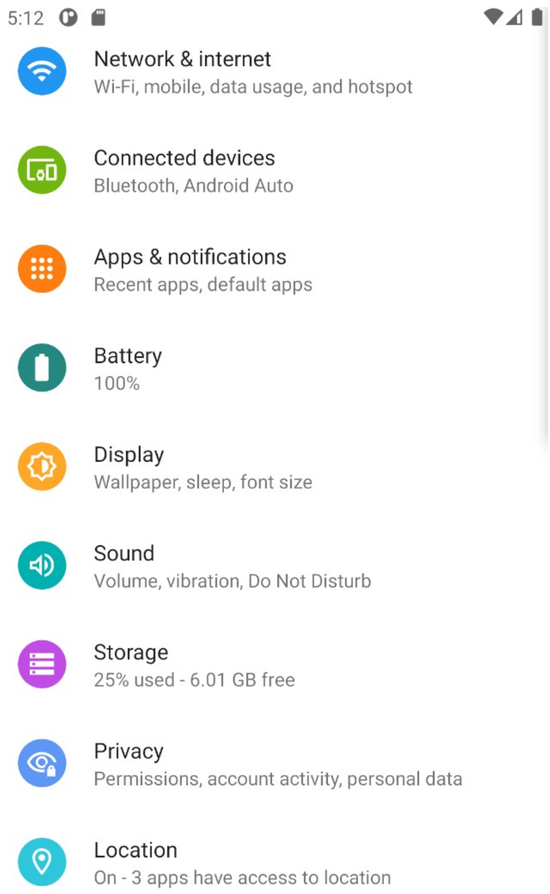
System Settings in Android



Google
Services & preferences

System
Languages, gestures, time, backup

About emulated device
sdk_gphone_x86_arm



Network & internet
Wi-Fi, mobile, data usage, and hotspot

Connected devices
Bluetooth, Android Auto

Apps & notifications
Recent apps, default apps

Battery
100%

Display
Wallpaper, sleep, font size

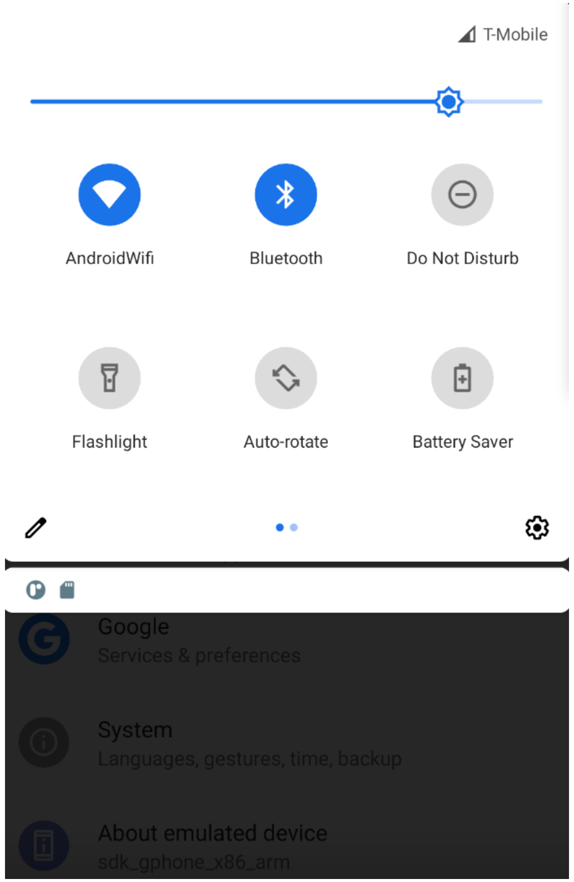
Sound
Volume, vibration, Do Not Disturb

Storage
25% used - 6.01 GB free

Privacy
Permissions, account activity, personal data

Location
On - 3 apps have access to location

System Settings in Android



T-Mobile

AndroidWifi

Bluetooth

Do Not Disturb

Flashlight

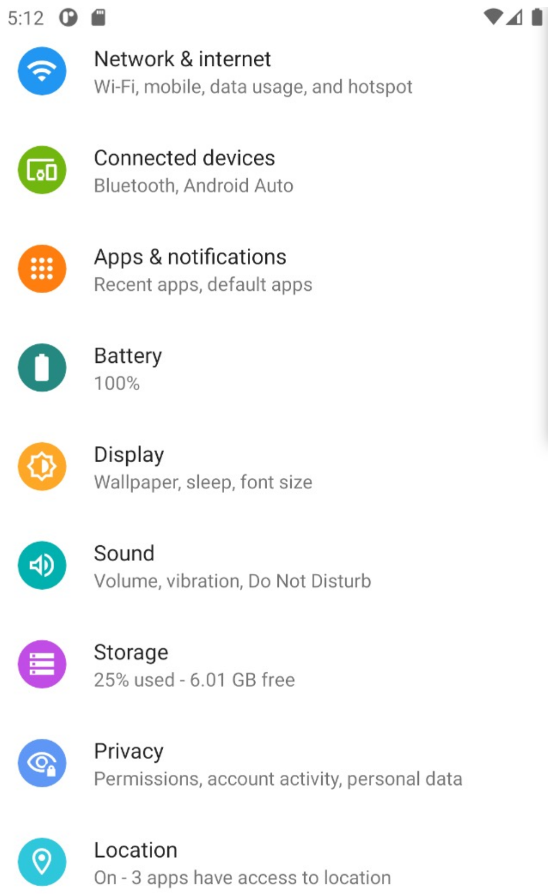
Auto-rotate

Battery Saver

Google
Services & preferences

System
Languages, gestures, time, backup

About emulated device
sdk_gphone_x86_arm



5:12

Network & internet
Wi-Fi, mobile, data usage, and hotspot

Connected devices
Bluetooth, Android Auto

Apps & notifications
Recent apps, default apps

Battery
100%

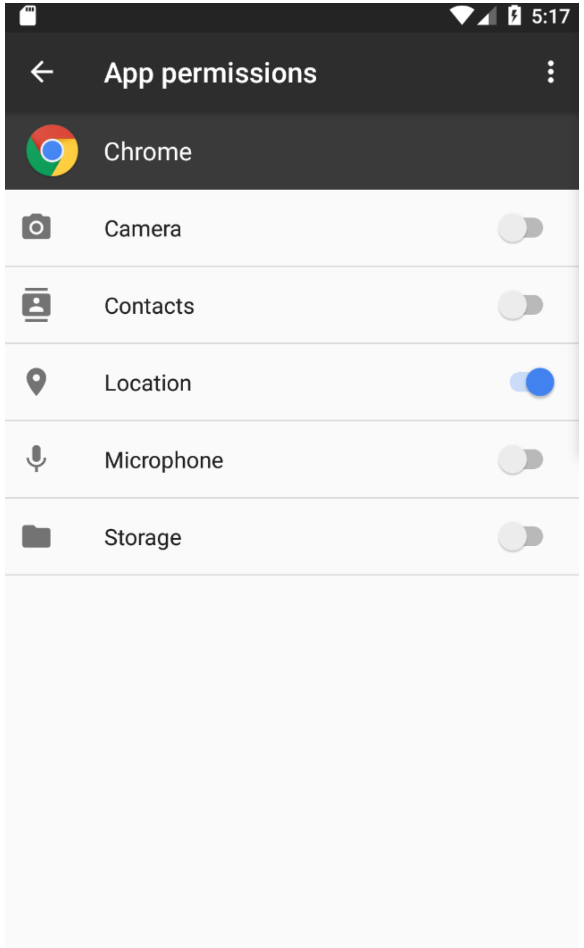
Display
Wallpaper, sleep, font size

Sound
Volume, vibration, Do Not Disturb

Storage
25% used - 6.01 GB free

Privacy
Permissions, account activity, personal data

Location
On - 3 apps have access to location



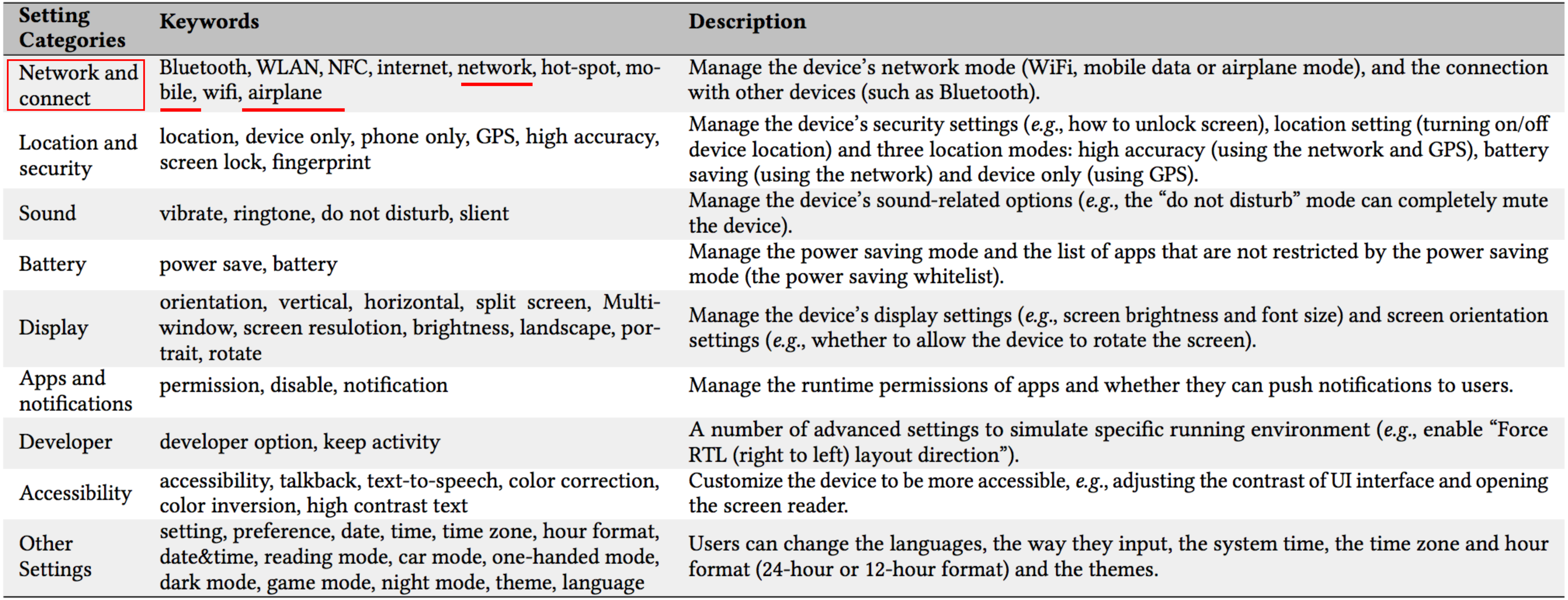
5:17

App permissions

Chrome

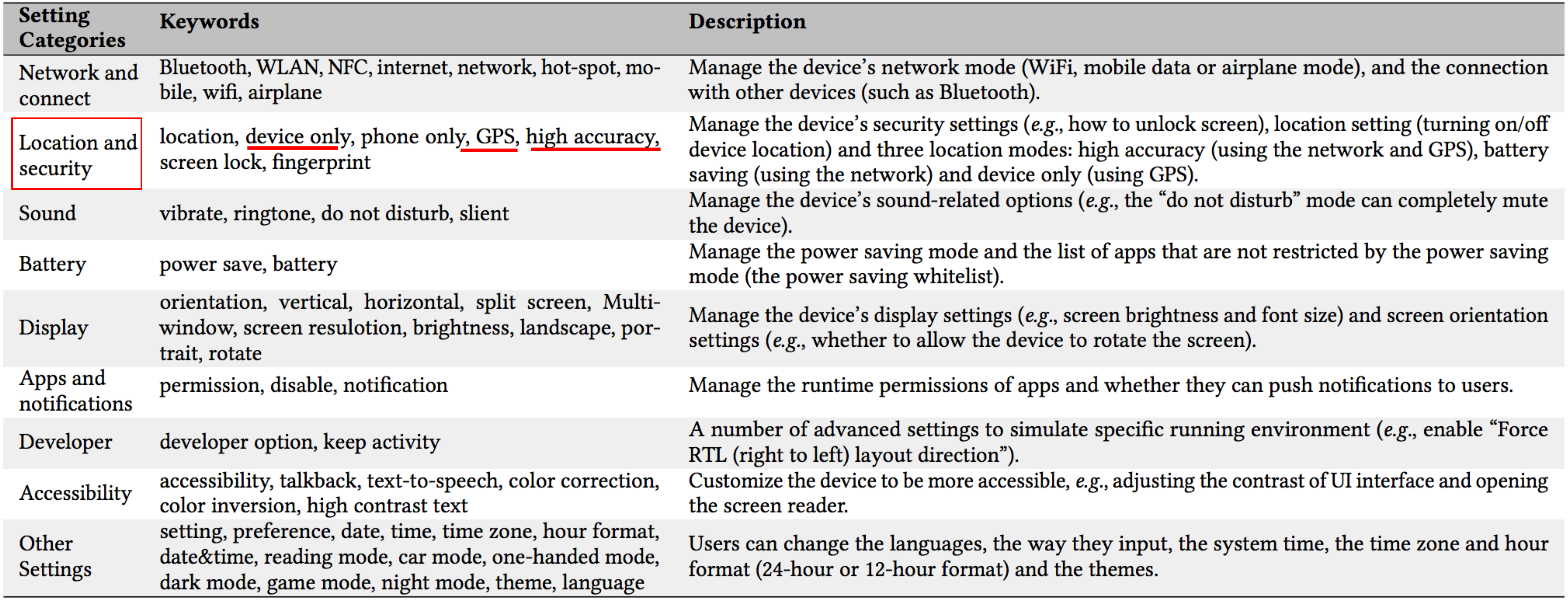
Camera	○
Contacts	○
Location	●
Microphone	○
Storage	○

System Settings in Android



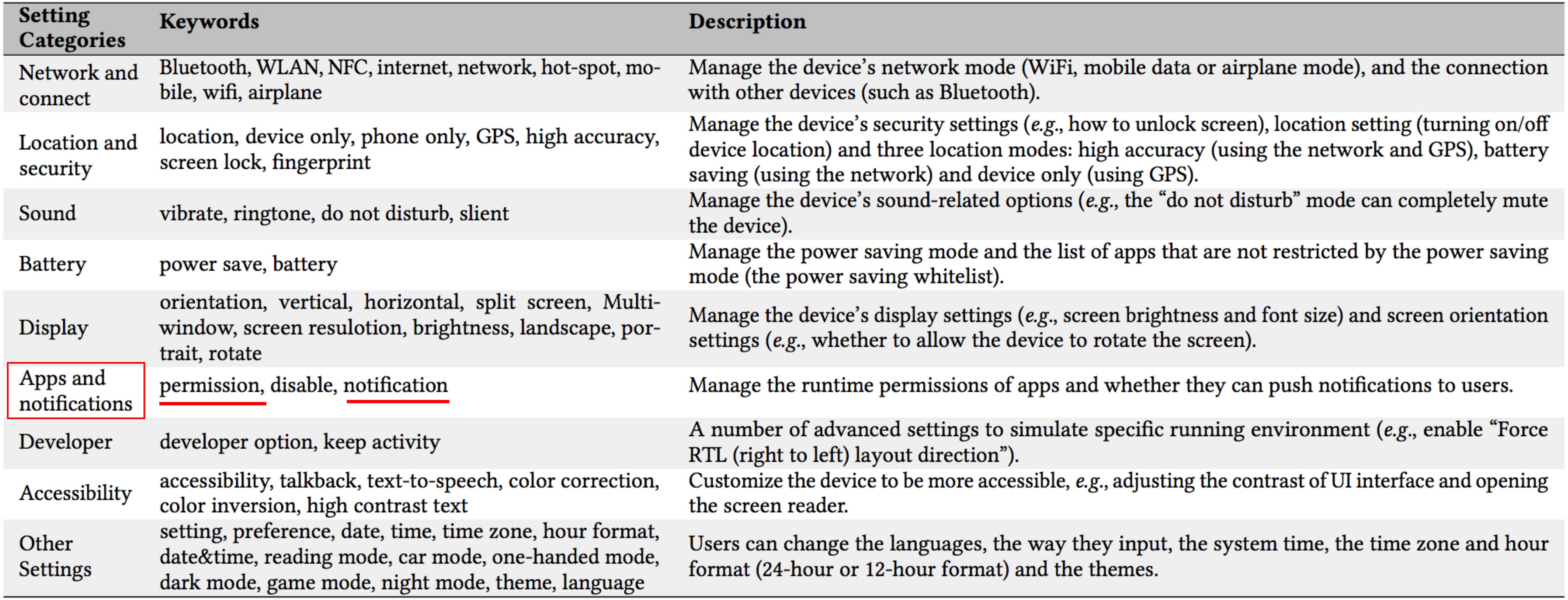
Setting Categories	Keywords	Description
Network and connect	Bluetooth, WLAN, NFC, internet, network, hot-spot, mobile, wifi, airplane	Manage the device's network mode (WiFi, mobile data or airplane mode), and the connection with other devices (such as Bluetooth).
Location and security	location, device only, phone only, GPS, high accuracy, screen lock, fingerprint	Manage the device's security settings (*e.g.*, how to unlock screen), location setting (turning on/off device location) and three location modes: high accuracy (using the network and GPS), battery saving (using the network) and device only (using GPS).
Sound	vibrate, ringtone, do not disturb, slient	Manage the device's sound-related options (*e.g.*, the "do not disturb" mode can completely mute the device).
Battery	power save, battery	Manage the power saving mode and the list of apps that are not restricted by the power saving mode (the power saving whitelist).
Display	orientation, vertical, horizontal, split screen, Multi-window, screen resulotion, brightness, landscape, portrait, rotate	Manage the device's display settings (*e.g.*, screen brightness and font size) and screen orientation settings (*e.g.*, whether to allow the device to rotate the screen).
Apps and notifications	permission, disable, notification	Manage the runtime permissions of apps and whether they can push notifications to users.
Developer	developer option, keep activity	A number of advanced settings to simulate specific running environment (*e.g.*, enable "Force RTL (right to left) layout direction").
Accessibility	accessibility, talkback, text-to-speech, color correction, color inversion, high contrast text	Customize the device to be more accessible, *e.g.*, adjusting the contrast of UI interface and opening the screen reader.
Other Settings	setting, preference, date, time, time zone, hour format, date&time, reading mode, car mode, one-handed mode, dark mode, game mode, night mode, theme, language	Users can change the languages, the way they input, the system time, the time zone and hour format (24-hour or 12-hour format) and the themes.

System Settings in Android



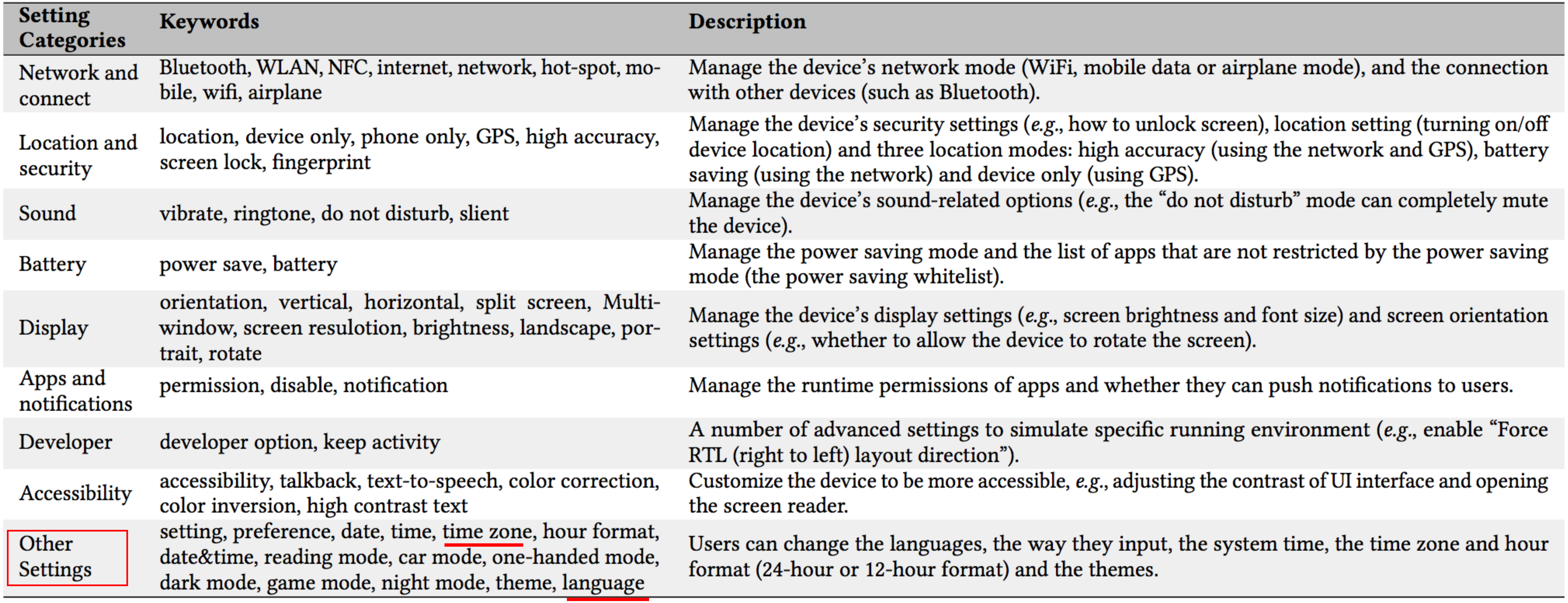
Setting Categories	Keywords	Description
Network and connect	Bluetooth, WLAN, NFC, internet, network, hot-spot, mobile, wifi, airplane	Manage the device's network mode (WiFi, mobile data or airplane mode), and the connection with other devices (such as Bluetooth).
Location and security	location, device only, phone only, GPS, high accuracy, screen lock, fingerprint	Manage the device's security settings (*e.g.*, how to unlock screen), location setting (turning on/off device location) and three location modes: high accuracy (using the network and GPS), battery saving (using the network) and device only (using GPS).
Sound	vibrate, ringtone, do not disturb, slient	Manage the device's sound-related options (*e.g.*, the "do not disturb" mode can completely mute the device).
Battery	power save, battery	Manage the power saving mode and the list of apps that are not restricted by the power saving mode (the power saving whitelist).
Display	orientation, vertical, horizontal, split screen, Multi-window, screen resulotion, brightness, landscape, portrait, rotate	Manage the device's display settings (*e.g.*, screen brightness and font size) and screen orientation settings (*e.g.*, whether to allow the device to rotate the screen).
Apps and notifications	permission, disable, notification	Manage the runtime permissions of apps and whether they can push notifications to users.
Developer	developer option, keep activity	A number of advanced settings to simulate specific running environment (*e.g.*, enable "Force RTL (right to left) layout direction").
Accessibility	accessibility, talkback, text-to-speech, color correction, color inversion, high contrast text	Customize the device to be more accessible, *e.g.*, adjusting the contrast of UI interface and opening the screen reader.
Other Settings	setting, preference, date, time, time zone, hour format, date&time, reading mode, car mode, one-handed mode, dark mode, game mode, night mode, theme, language	Users can change the languages, the way they input, the system time, the time zone and hour format (24-hour or 12-hour format) and the themes.

System Settings in Android

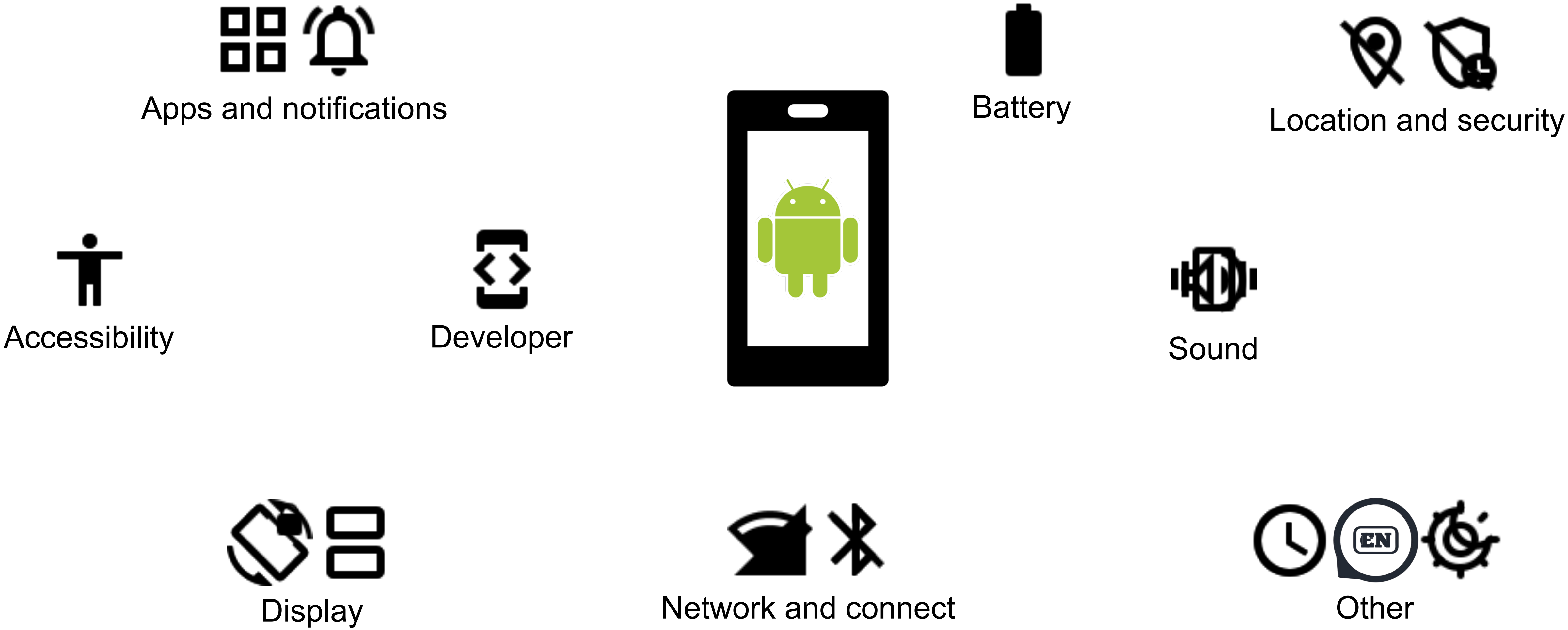


Setting Categories	Keywords	Description
Network and connect	Bluetooth, WLAN, NFC, internet, network, hot-spot, mobile, wifi, airplane	Manage the device's network mode (WiFi, mobile data or airplane mode), and the connection with other devices (such as Bluetooth).
Location and security	location, device only, phone only, GPS, high accuracy, screen lock, fingerprint	Manage the device's security settings (*e.g.*, how to unlock screen), location setting (turning on/off device location) and three location modes: high accuracy (using the network and GPS), battery saving (using the network) and device only (using GPS).
Sound	vibrate, ringtone, do not disturb, slient	Manage the device's sound-related options (*e.g.*, the "do not disturb" mode can completely mute the device).
Battery	power save, battery	Manage the power saving mode and the list of apps that are not restricted by the power saving mode (the power saving whitelist).
Display	orientation, vertical, horizontal, split screen, Multi-window, screen resulotion, brightness, landscape, portrait, rotate	Manage the device's display settings (*e.g.*, screen brightness and font size) and screen orientation settings (*e.g.*, whether to allow the device to rotate the screen).
Apps and notifications	permission, disable, notification	Manage the runtime permissions of apps and whether they can push notifications to users.
Developer	developer option, keep activity	A number of advanced settings to simulate specific running environment (*e.g.*, enable "Force RTL (right to left) layout direction").
Accessibility	accessibility, talkback, text-to-speech, color correction, color inversion, high contrast text	Customize the device to be more accessible, *e.g.*, adjusting the contrast of UI interface and opening the screen reader.
Other Settings	setting, preference, date, time, time zone, hour format, date&time, reading mode, car mode, one-handed mode, dark mode, game mode, night mode, theme, language	Users can change the languages, the way they input, the system time, the time zone and hour format (24-hour or 12-hour format) and the themes.

System Settings in Android



Setting Categories	Keywords	Description
Network and connect	Bluetooth, WLAN, NFC, internet, network, hot-spot, mobile, wifi, airplane	Manage the device's network mode (WiFi, mobile data or airplane mode), and the connection with other devices (such as Bluetooth).
Location and security	location, device only, phone only, GPS, high accuracy, screen lock, fingerprint	Manage the device's security settings (*e.g.*, how to unlock screen), location setting (turning on/off device location) and three location modes: high accuracy (using the network and GPS), battery saving (using the network) and device only (using GPS).
Sound	vibrate, ringtone, do not disturb, slient	Manage the device's sound-related options (*e.g.*, the "do not disturb" mode can completely mute the device).
Battery	power save, battery	Manage the power saving mode and the list of apps that are not restricted by the power saving mode (the power saving whitelist).
Display	orientation, vertical, horizontal, split screen, Multi-window, screen resulotion, brightness, landscape, portrait, rotate	Manage the device's display settings (*e.g.*, screen brightness and font size) and screen orientation settings (*e.g.*, whether to allow the device to rotate the screen).
Apps and notifications	permission, disable, notification	Manage the runtime permissions of apps and whether they can push notifications to users.
Developer	developer option, keep activity	A number of advanced settings to simulate specific running environment (*e.g.*, enable "Force RTL (right to left) layout direction").
Accessibility	accessibility, talkback, text-to-speech, color correction, color inversion, high contrast text	Customize the device to be more accessible, *e.g.*, adjusting the contrast of UI interface and opening the screen reader.
Other Settings	setting, preference, date, time, time zone, hour format, date&time, reading mode, car mode, one-handed mode, dark mode, game mode, night mode, theme, language	Users can change the languages, the way they input, the system time, the time zone and hour format (24-hour or 12-hour format) and the themes.



Apps and notifications

Battery

Location and security

Accessibility

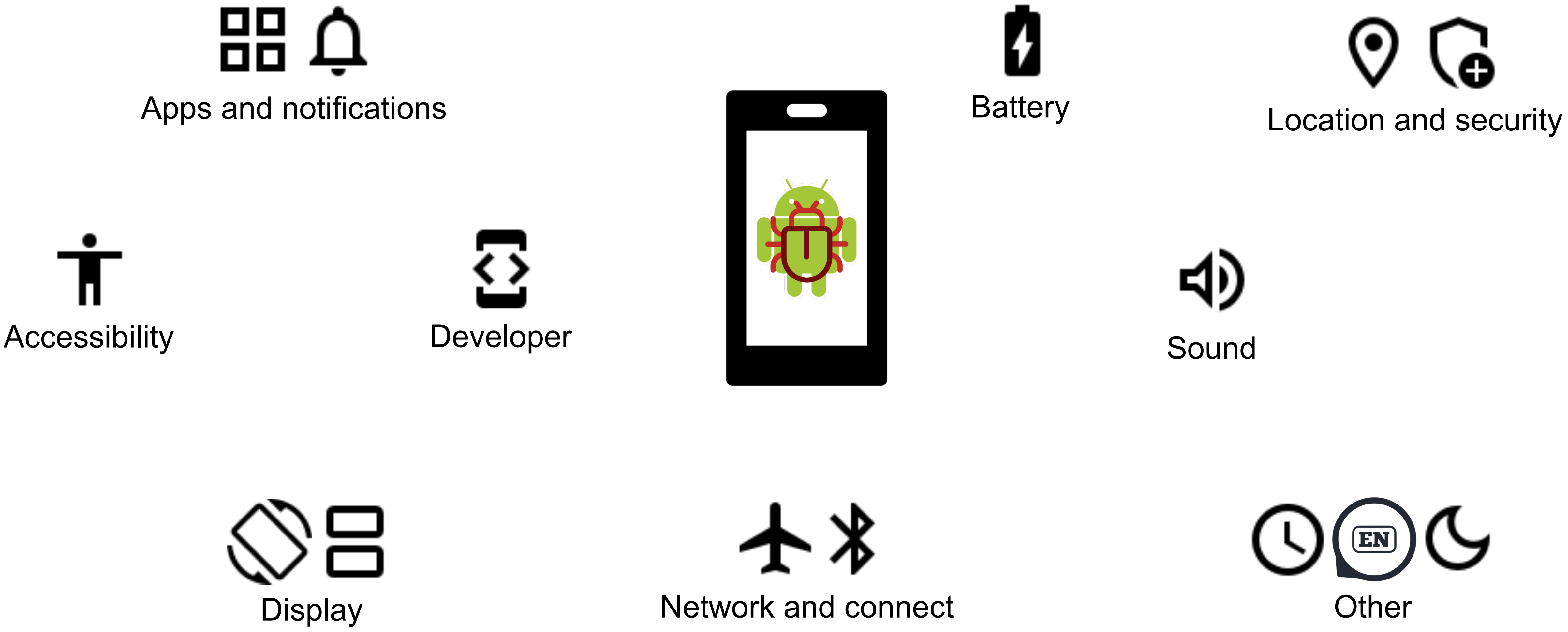
Developer

Sound

Display

Network and connect

Other



Apps and notifications

Battery

Location and security

Accessibility

Developer

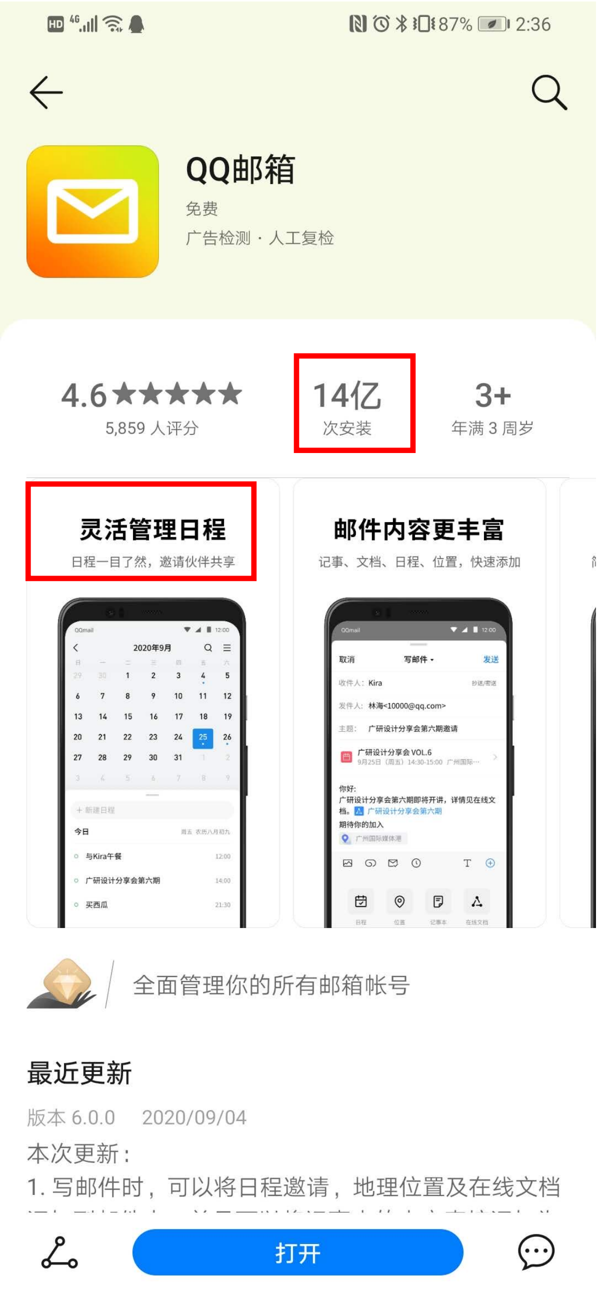
Sound

Display

Network and connect

Other

A QQ-Mail Bug



87% 2:36

QQ邮箱

免费

广告检测 · 人工复检

4.6★★★★★
5,859 人评分

14亿
次安装

3+
年满 3 周岁

灵活管理日程
日程一目了然，邀请伙伴共享

邮件内容更丰富
记事、文档、日程、位置，快速添加

全面管理你的所有邮箱帐号

最近更新

版本 6.0.0 2020/09/04

本次更新：

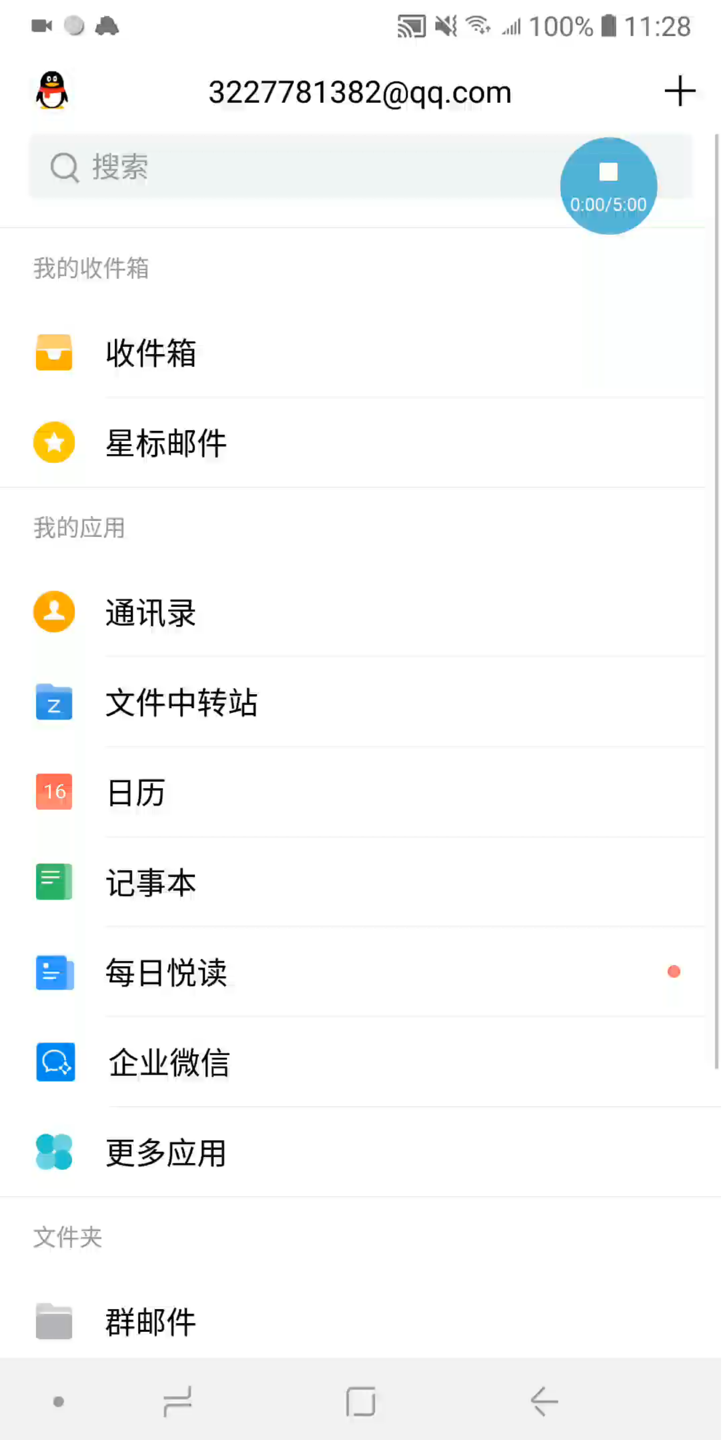
1. 写邮件时，可以将日程邀请，地理位置及在线文档

打开

- **Consequence:**
  cannot delete a to-do task

- **Root cause:**
  Fail to properly handle the calendar permission

- **Status:**
  Confirmed and fixed



100% 11:28

3227781382@qq.com

搜索

0:00/5:00

我的收件箱

收件箱

星标邮件

我的应用

通讯录

文件中转站

日历

记事本

每日悦读

企业微信

更多应用

文件夹

群邮件

Consequences of System Setting-related Functional Bugs

Consequence Type	Description	#Defects
Crash	App crashes	315
Disrespect of setting changes	Do not reflect setting changes, e.g., untranslated texts or incomplete translations when the system language is changed.	218
Problematic UI display	GUI display issues	197
Function failure	Function cannot work, e.g., app stuck, black screen, infinite loading, and unable to refresh.	197
Total		**1074**

* We analyzed 1074 setting issues from 180 popular, well-maintained Android apps on GitHub

Consequences of System Setting-related Functional Bugs

Consequence Type	Description	#Defects
Crash	App crashes	315
Disrespect of setting changes	Do not reflect setting changes, e.g., untranslated texts or incomplete translations when the system language is changed.	218
Problematic UI display	GUI display issues	197
Function failure	Function cannot work, e.g., app stuck, black screen, infinite loading, and unable to refresh.	197
Total		**1074**

functional bugs

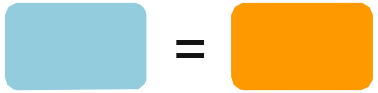
* We analyzed 1074 setting issues from 180 popular, well-maintained Android apps on GitHub

⇒ 70.7% of setting defects are non-crashing functional bugs

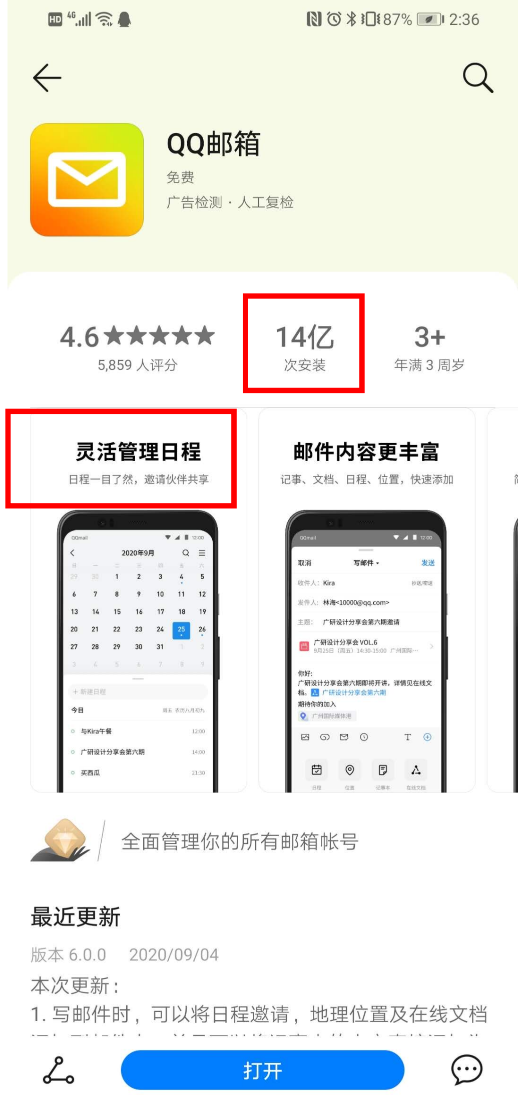
How to use *metamorphic testing* to find such system setting-related *non-crashing* functional bugs?

Metamorphic Relations for Setting Defects

- MR1: If a given system setting is changed *and later* restored,
  ⇒ an app's behaviors (functionalities) should *keep consistent*



A QQ-Mail Bug



87% 2:36

QQ邮箱

免费

广告检测 · 人工复检

4.6★★★★★
5,859 人评分

14亿
次安装

3+
年满 3 周岁

灵活管理日程
日程一目了然，邀请伙伴共享

邮件内容更丰富
记事、文档、日程、位置，快速添加

全面管理你的所有邮箱帐号

最近更新

版本 6.0.0 2020/09/04

本次更新：

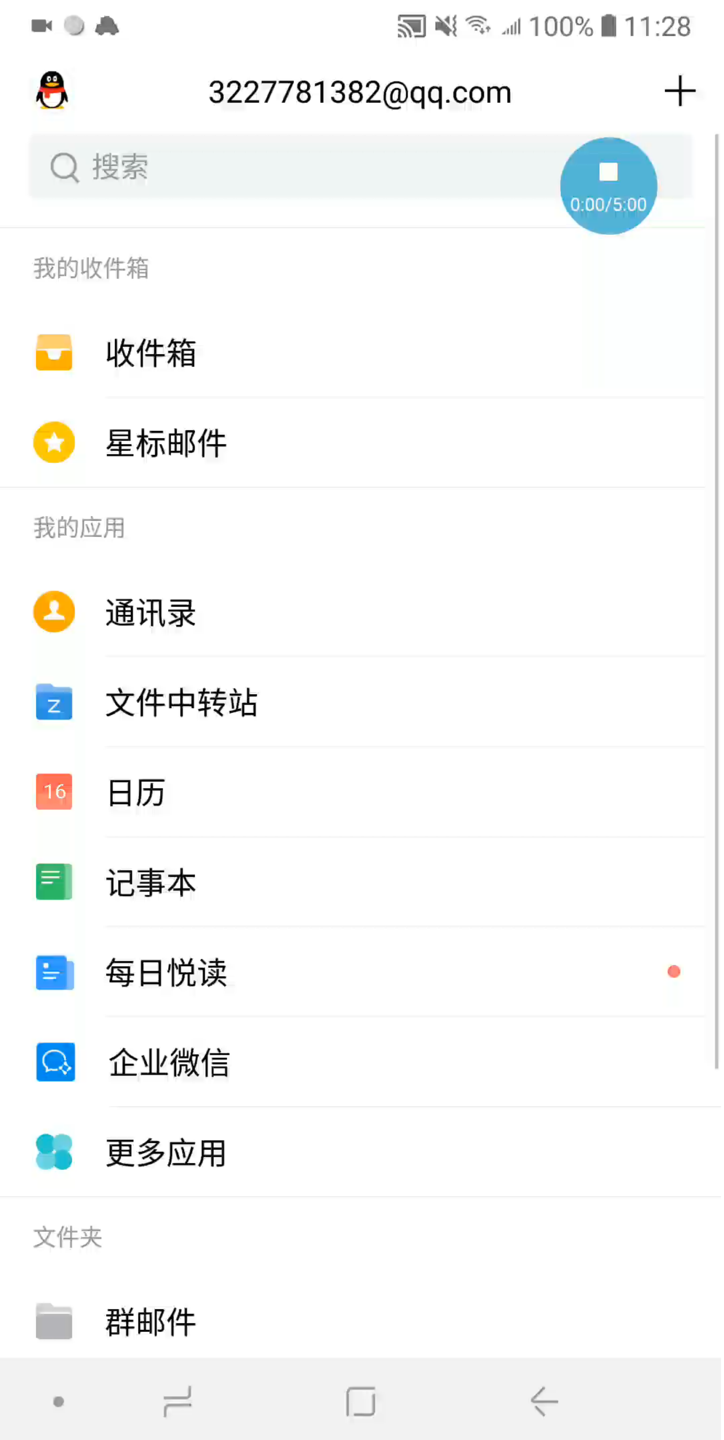
1. 写邮件时，可以将日程邀请，地理位置及在线文档

打开

- **Consequence:**
  - cannot delete a to-do task

- **Root cause:**
  - Fail to properly handle the calendar permission

- **Status:**
  - Confirmed and fixed



100% 11:28

3227781382@qq.com

搜索

0:00/5:00

我的收件箱

收件箱

星标邮件

我的应用

通讯录

文件中转站

日历

记事本

每日悦读

企业微信

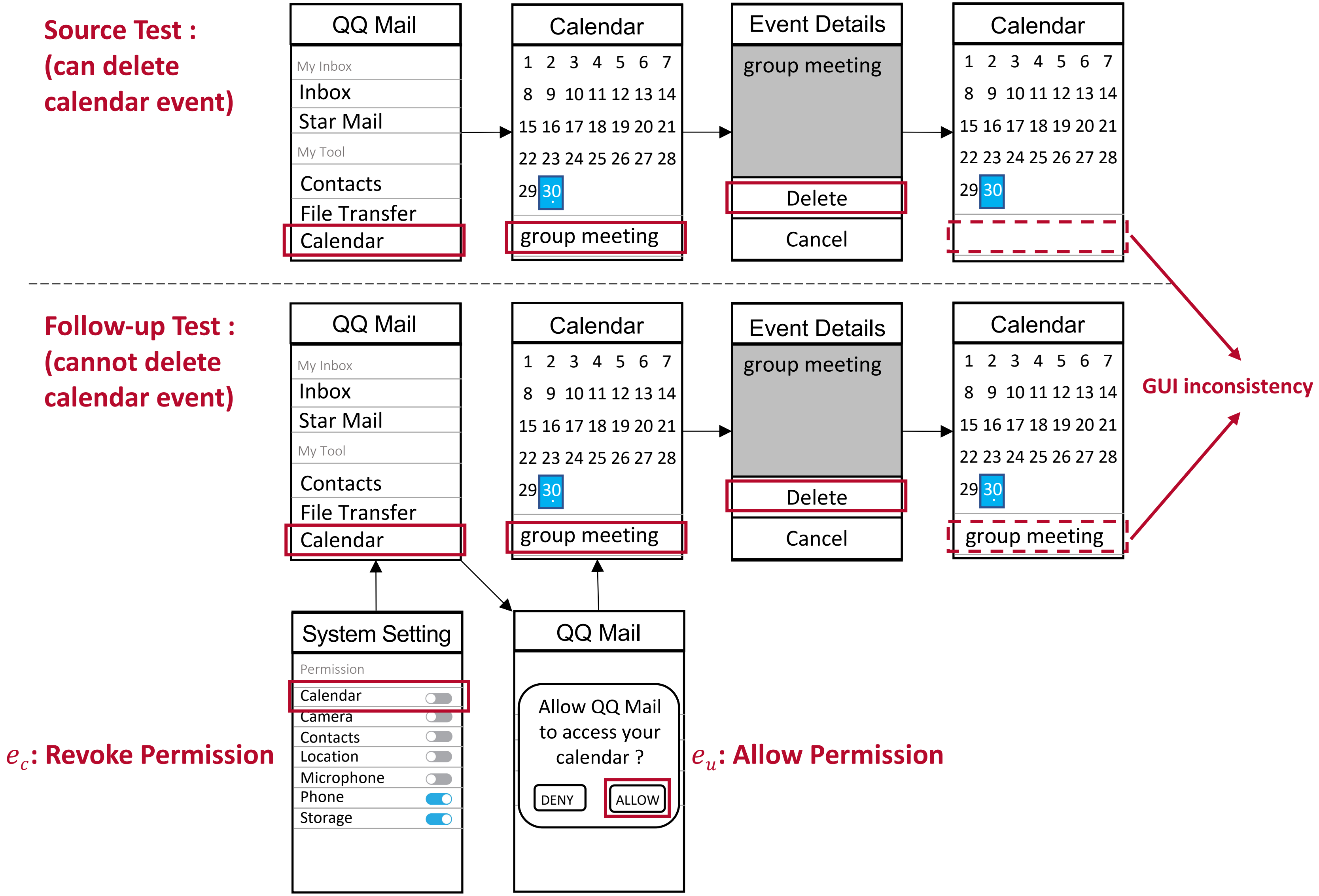
更多应用

文件夹

群邮件

QQ Mail

4,500,000,000+
installations in major
app markets



Source Test :
(can delete calendar event)

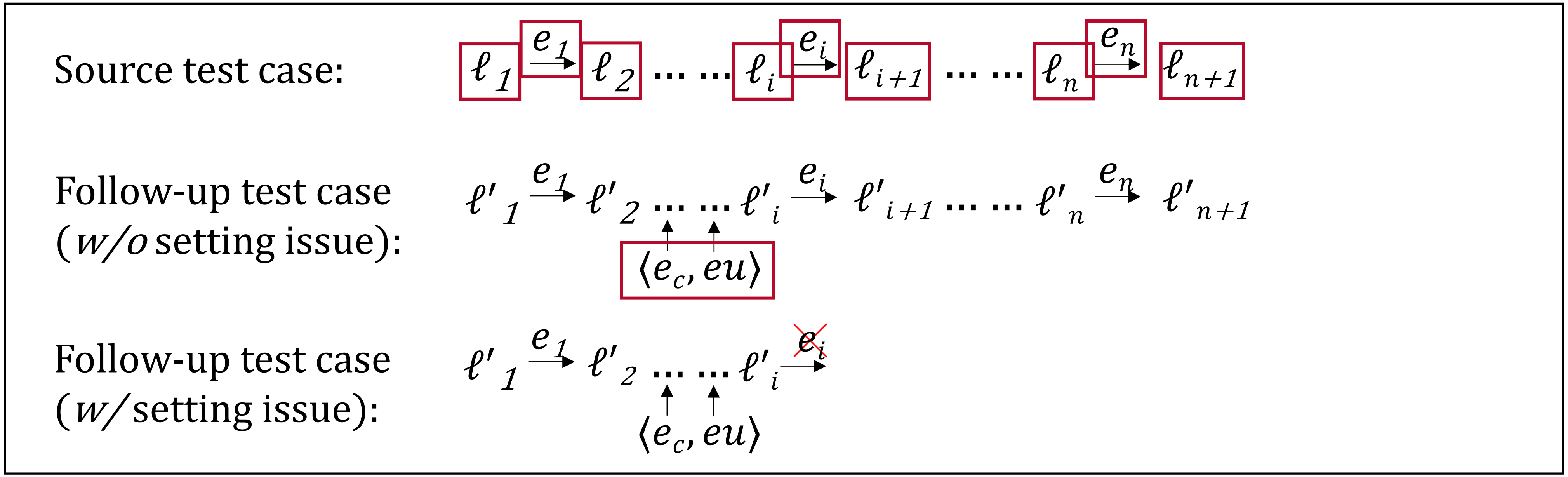
Follow-up Test :
(cannot delete calendar event)

e_c: **Revoke Permission**

e_u: **Allow Permission**

GUI inconsistency

Our Approach



Source test case: $\boxed{\ell_1}\boxed{\xrightarrow{e_1}}\boxed{\ell_2} \dots \dots \boxed{\ell_i}\boxed{\xrightarrow{e_i}}\boxed{\ell_{i+1}} \dots \dots \boxed{\ell_n}\boxed{\xrightarrow{e_n}}\boxed{\ell_{n+1}}$

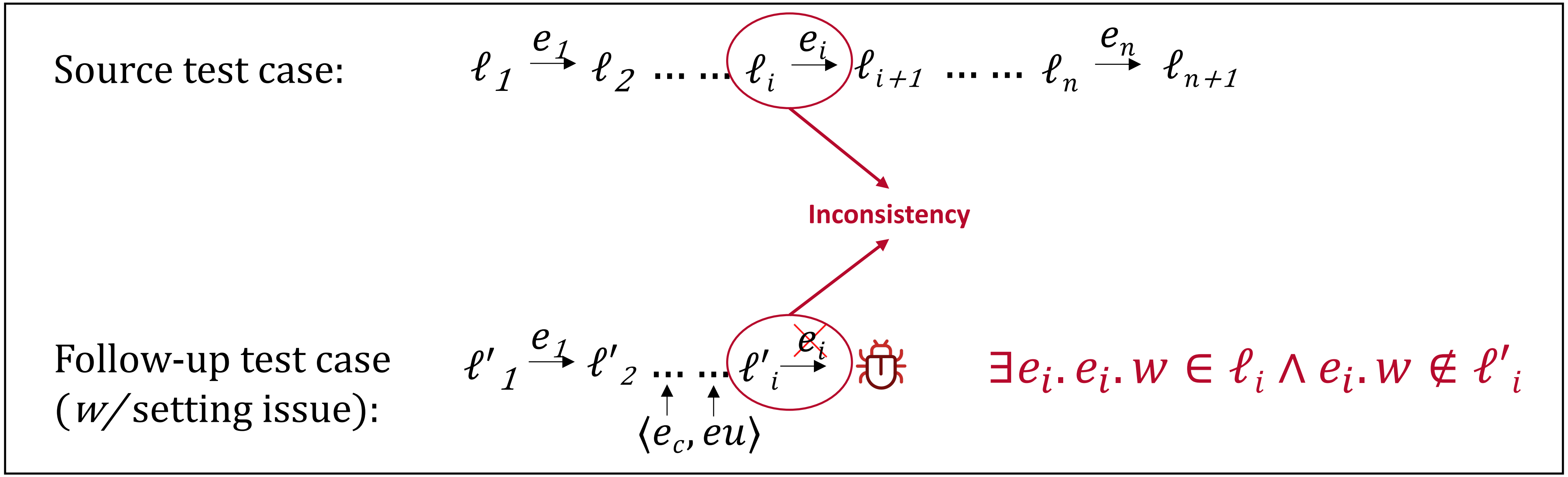
Follow-up test case (*w/o* setting issue): $\ell'_1 \xrightarrow{e_1} \ell'_2 \underset{\boxed{\langle e_c, eu \rangle}}{\underset{\uparrow \quad \uparrow}{\dots \dots}} \ell'_i \xrightarrow{e_i} \ell'_{i+1} \dots \dots \ell'_n \xrightarrow{e_n} \ell'_{n+1}$

Follow-up test case (*w/* setting issue): $\ell'_1 \xrightarrow{e_1} \ell'_2 \underset{\langle e_c, eu \rangle}{\underset{\uparrow \quad \uparrow}{\dots \dots}} \ell'_i \xrightarrow{\cancel{e_i}}$

e — a GUI event $\qquad$ e_c — changes a given setting

ℓ — a GUI layout $\qquad$ e_u — restores the setting.

Our Approach



Source test case: $\ell_1 \xrightarrow{e_1} \ell_2 \dots\dots \ell_i \xrightarrow{e_i} \ell_{i+1} \dots\dots \ell_n \xrightarrow{e_n} \ell_{n+1}$

Follow-up test case ($w/$setting issue): $\ell'_1 \xrightarrow{e_1} \ell'_2 \dots\dots \ell'_i \xrightarrow{\cancel{e_i}}$

$\langle e_c, eu \rangle$

$\exists e_i . e_i . w \in \ell_i \wedge e_i . w \notin \ell'_i$

e — a GUI event

ℓ — a GUI layout

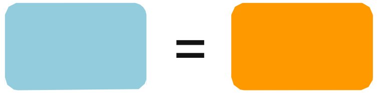
e_c — changes a given setting

e_u — restores the setting.

$e.w$ — target widget of GUI event

Metamorphic Relations for Setting Defects

- MR1: If a given system setting is changed *and later* restored,
  ⇒ an app's behaviors (functionalities) should *keep consistent*



Metamorphic Relations for Setting Defects

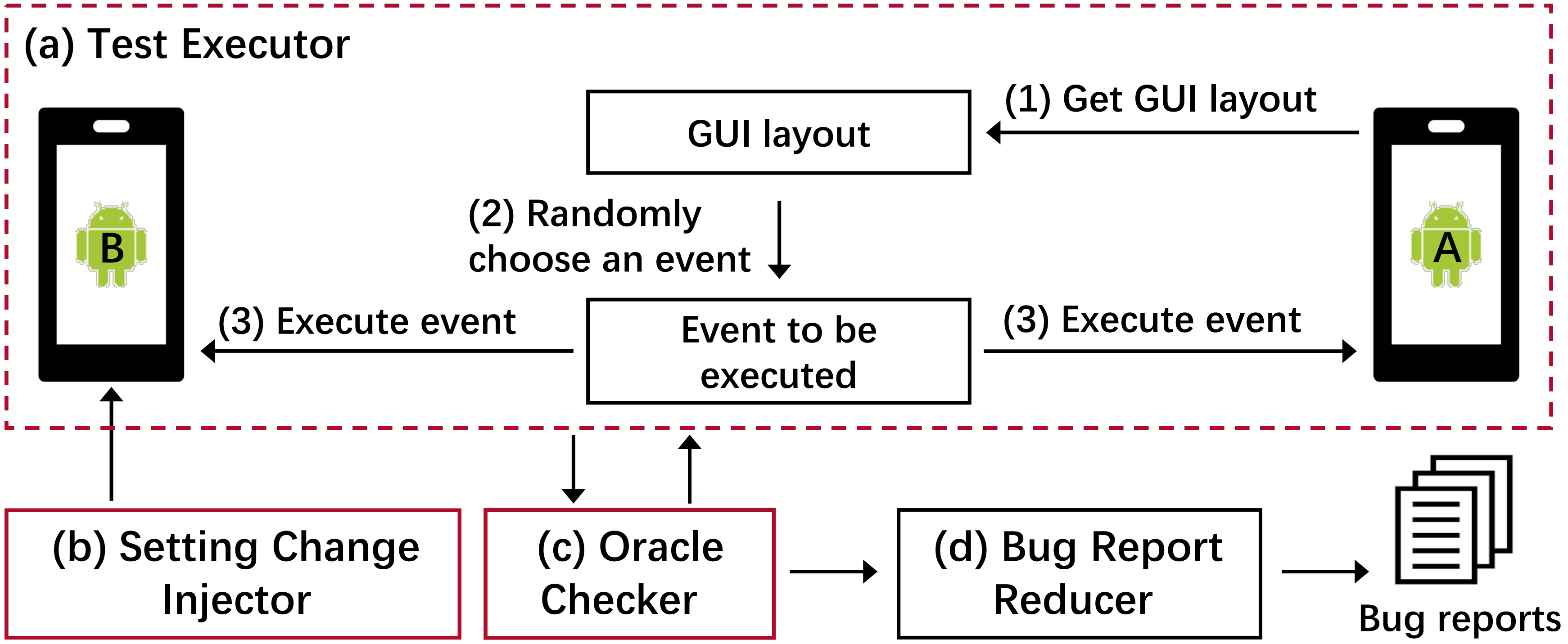
- MR1: If a given system setting is changed *and later* restored,
  ⇒ an app’s behaviors (functionalities) should *keep consistent*

🟦 = 🟧

- MR2: If a given system setting is changed *but not* restored
  ⇒ an app’s behaviors (functionalities) should *show differences*

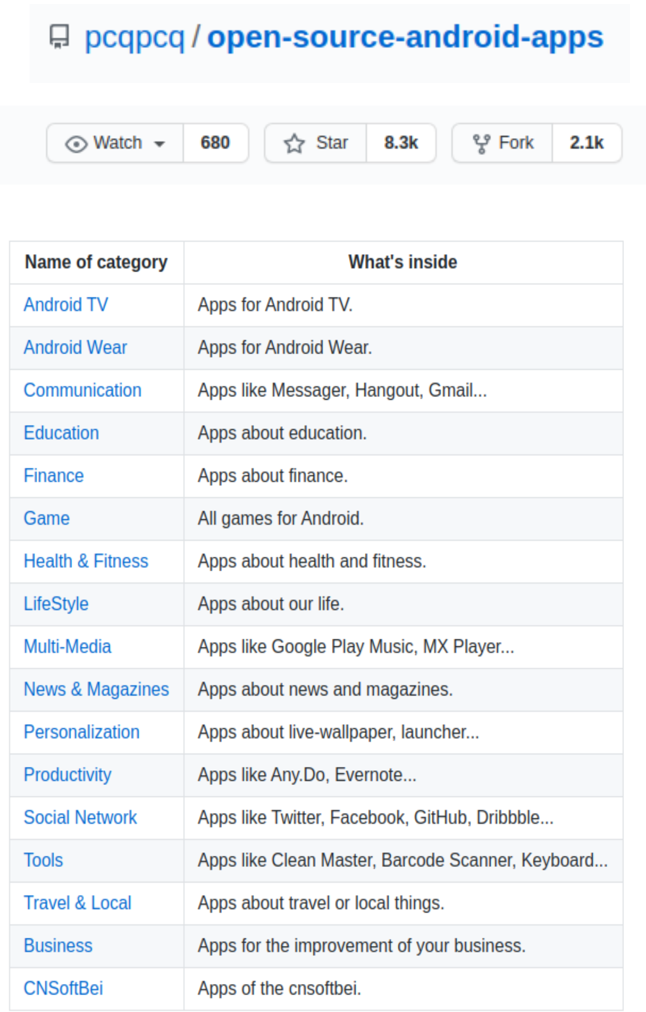
🟦 ≠ 🟧

SetDroid: Setting-wise Metamorphic Fuzzing



Workflow of Our Approach

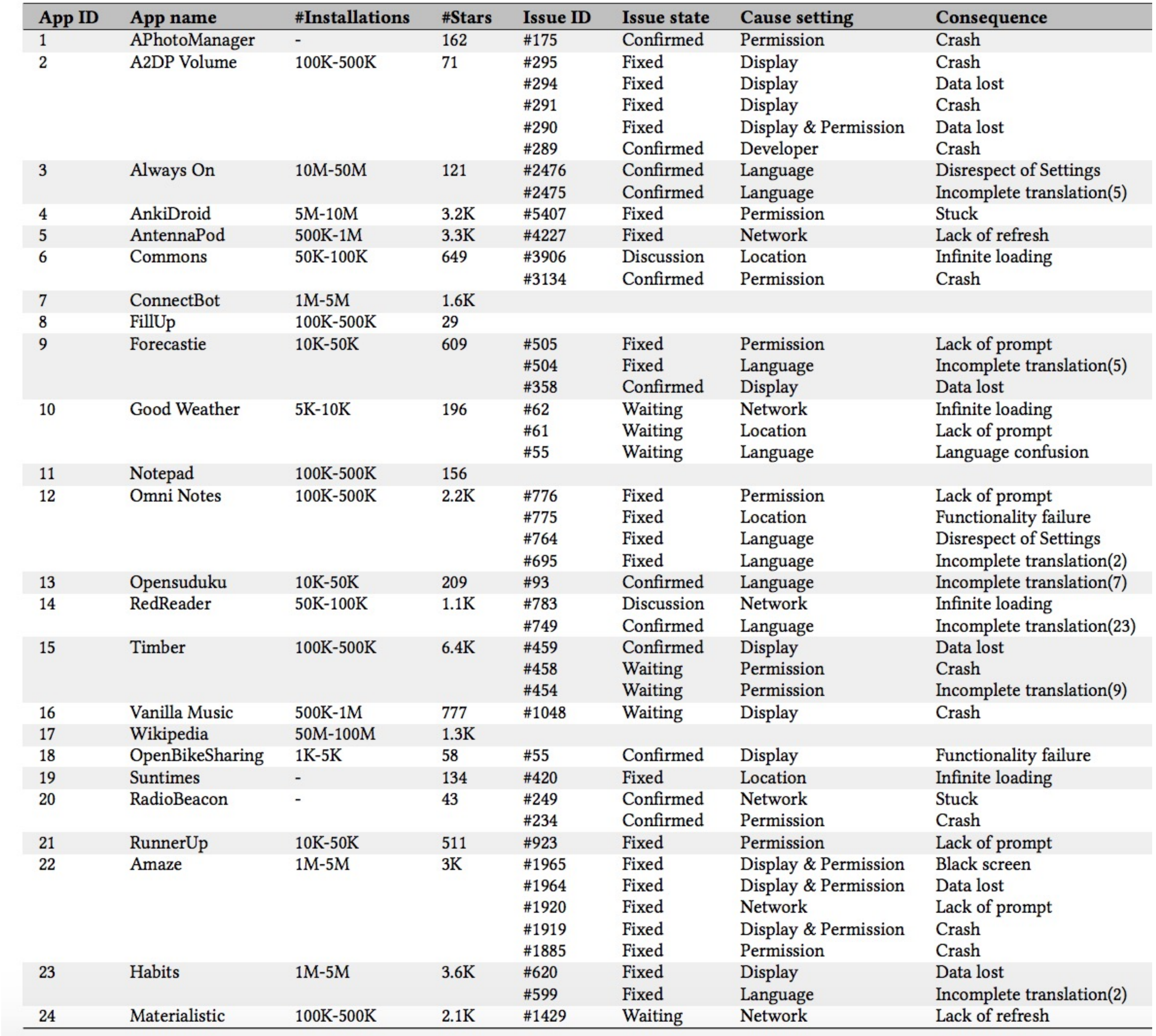
Results: Bug Finding in Open-source Apps



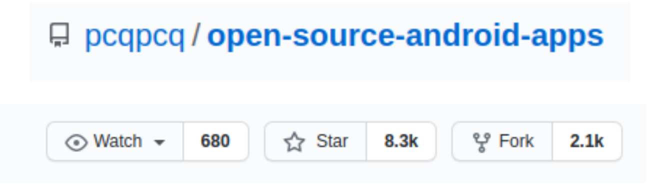
pcqpcq / open-source-android-apps

Name of category	What's inside
Android TV	Apps for Android TV.
Android Wear	Apps for Android Wear.
Communication	Apps like Messager, Hangout, Gmail...
Education	Apps about education.
Finance	Apps about finance.
Game	All games for Android.
Health & Fitness	Apps about health and fitness.
LifeStyle	Apps about our life.
Multi-Media	Apps like Google Play Music, MX Player...
News & Magazines	Apps about news and magazines.
Personalization	Apps about live-wallpaper, launcher...
Productivity	Apps like Any.Do, Evernote...
Social Network	Apps like Twitter, Facebook, GitHub, Dribbble...
Tools	Apps like Clean Master, Barcode Scanner, Keyboard...
Travel & Local	Apps about travel or local things.
Business	Apps for the improvement of your business.
CNSoftBei	Apps of the cnsoftbei.

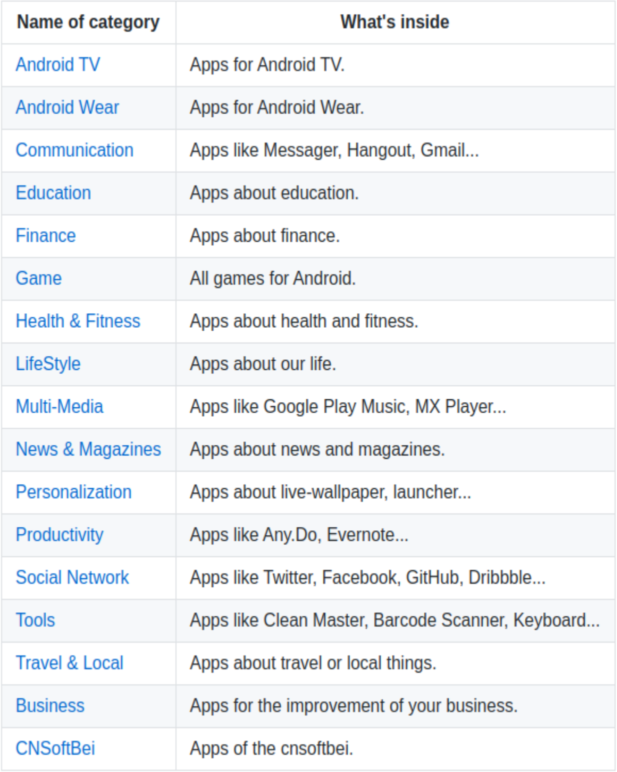
- ✓ **Selected 26 apps**
- ✓ **Detected 42 bugs in 24 apps**



App ID	App name	#Installations	#Stars	Issue ID	Issue state	Cause setting	Consequence
1	APhotoManager	-	162	#175	Confirmed	Permission	Crash
2	A2DP Volume	100K-500K	71	#295	Fixed	Display	Crash
				#294	Fixed	Display	Data lost
				#291	Fixed	Display	Crash
				#290	Fixed	Display & Permission	Data lost
				#289	Confirmed	Developer	Crash
3	Always On	10M-50M	121	#2476	Confirmed	Language	Disrespect of Settings
				#2475	Confirmed	Language	Incomplete translation(5)
4	AnkiDroid	5M-10M	3.2K	#5407	Fixed	Permission	Stuck
5	AntennaPod	500K-1M	3.3K	#4227	Fixed	Network	Lack of refresh
6	Commons	50K-100K	649	#3906	Discussion	Location	Infinite loading
				#3134	Confirmed	Permission	Crash
7	ConnectBot	1M-5M	1.6K				
8	FillUp	100K-500K	29				
9	Forecastie	10K-50K	609	#505	Fixed	Permission	Lack of prompt
				#504	Fixed	Language	Incomplete translation(5)
				#358	Confirmed	Display	Data lost
10	Good Weather	5K-10K	196	#62	Waiting	Network	Infinite loading
				#61	Waiting	Location	Lack of prompt
				#55	Waiting	Language	Language confusion
11	Notepad	100K-500K	156				
12	Omni Notes	100K-500K	2.2K	#776	Fixed	Permission	Lack of prompt
				#775	Fixed	Location	Functionality failure
				#764	Fixed	Language	Disrespect of Settings
				#695	Fixed	Language	Incomplete translation(2)
13	Opensuduku	10K-50K	209	#93	Confirmed	Language	Incomplete translation(7)
14	RedReader	50K-100K	1.1K	#783	Discussion	Network	Infinite loading
				#749	Confirmed	Language	Incomplete translation(23)
15	Timber	100K-500K	6.4K	#459	Confirmed	Display	Data lost
				#458	Waiting	Permission	Crash
				#454	Waiting	Permission	Incomplete translation(9)
16	Vanilla Music	500K-1M	777	#1048	Waiting	Display	Crash
17	Wikipedia	50M-100M	1.3K				
18	OpenBikeSharing	1K-5K	58	#55	Confirmed	Display	Functionality failure
19	Suntimes	-	134	#420	Fixed	Location	Infinite loading
20	RadioBeacon	-	43	#249	Confirmed	Network	Stuck
				#234	Confirmed	Permission	Crash
21	RunnerUp	10K-50K	511	#923	Fixed	Permission	Lack of prompt
22	Amaze	1M-5M	3K	#1965	Fixed	Display & Permission	Black screen
				#1964	Fixed	Display & Permission	Data lost
				#1920	Fixed	Network	Lack of prompt
				#1919	Fixed	Display & Permission	Crash
				#1885	Fixed	Permission	Crash
23	Habits	1M-5M	3.6K	#620	Fixed	Display	Data lost
				#599	Fixed	Language	Incomplete translation(2)
24	Materialistic	100K-500K	2.1K	#1429	Waiting	Network	Lack of refresh

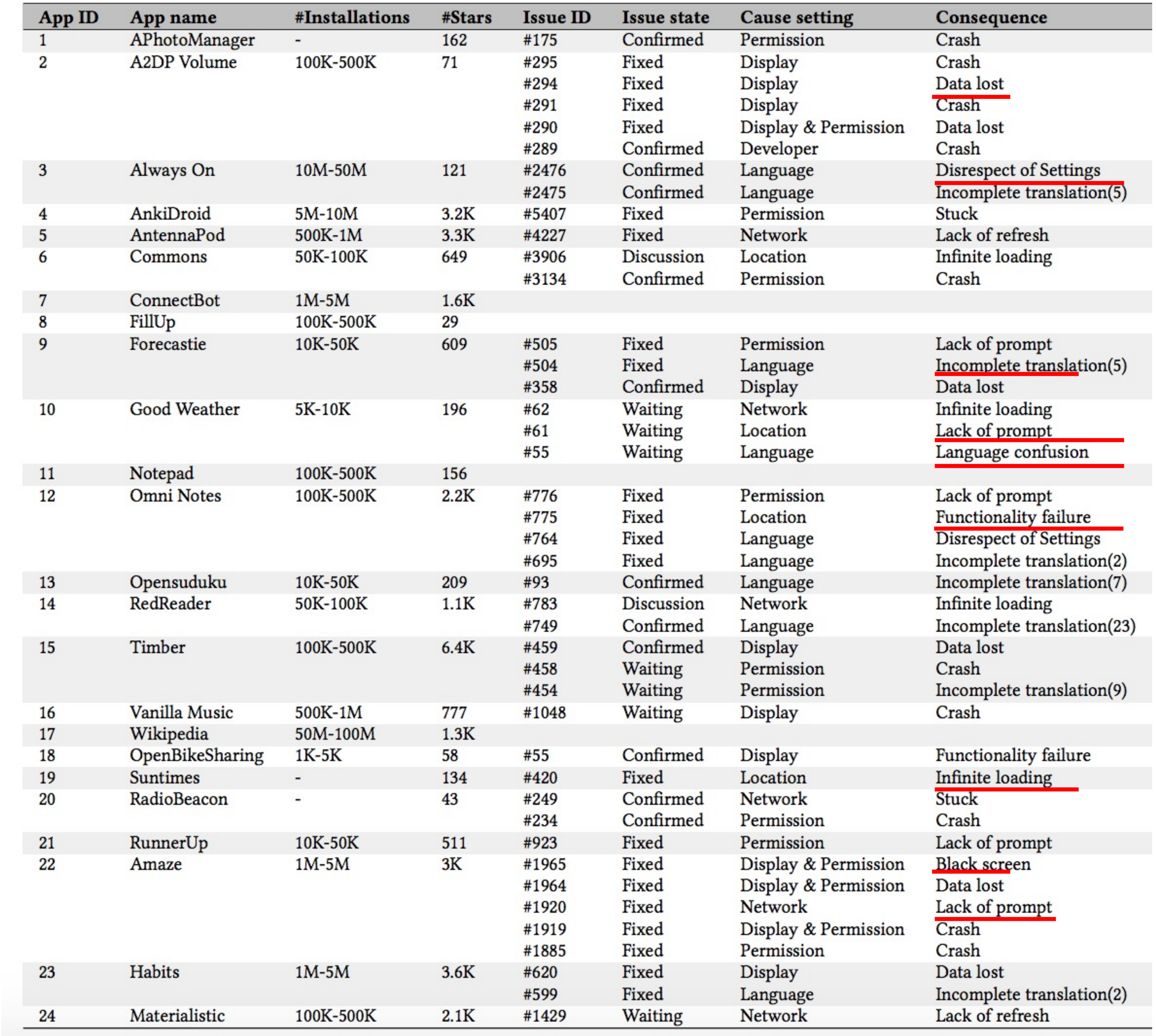


pcqpcq / open-source-android-apps



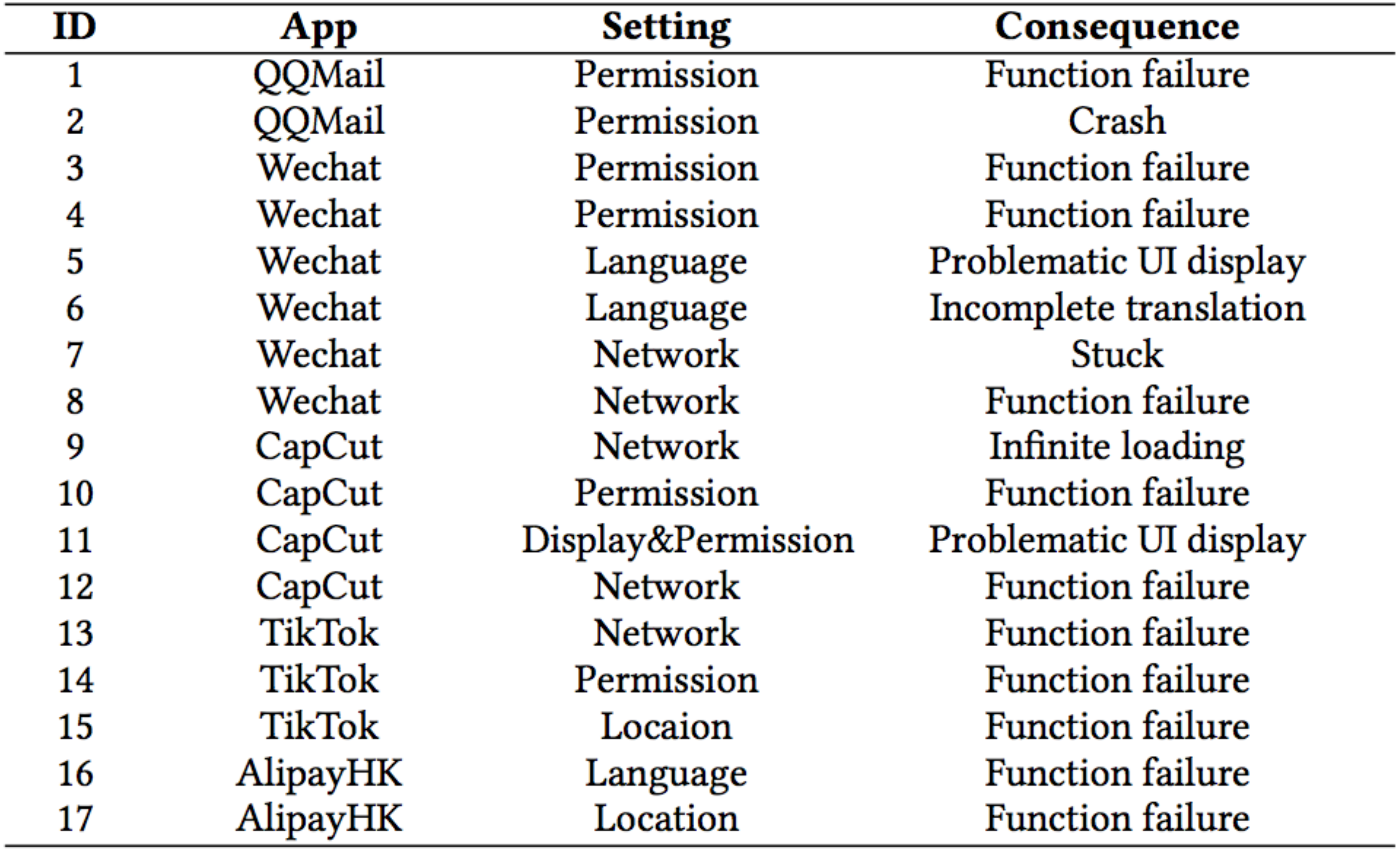
Name of category	What's inside
Android TV	Apps for Android TV.
Android Wear	Apps for Android Wear.
Communication	Apps like Messager, Hangout, Gmail...
Education	Apps about education.
Finance	Apps about finance.
Game	All games for Android.
Health & Fitness	Apps about health and fitness.
LifeStyle	Apps about our life.
Multi-Media	Apps like Google Play Music, MX Player...
News & Magazines	Apps about news and magazines.
Personalization	Apps about live-wallpaper, launcher...
Productivity	Apps like Any.Do, Evernote...
Social Network	Apps like Twitter, Facebook, GitHub, Dribbble...
Tools	Apps like Clean Master, Barcode Scanner, Keyboard...
Travel & Local	Apps about travel or local things.
Business	Apps for the improvement of your business.
CNSoftBei	Apps of the cnsoftbei.

- ✓ **Selected 26 apps**
- ✓ **Detected 42 bugs in 24 apps**
- ✓ **32/42 are functional bugs**



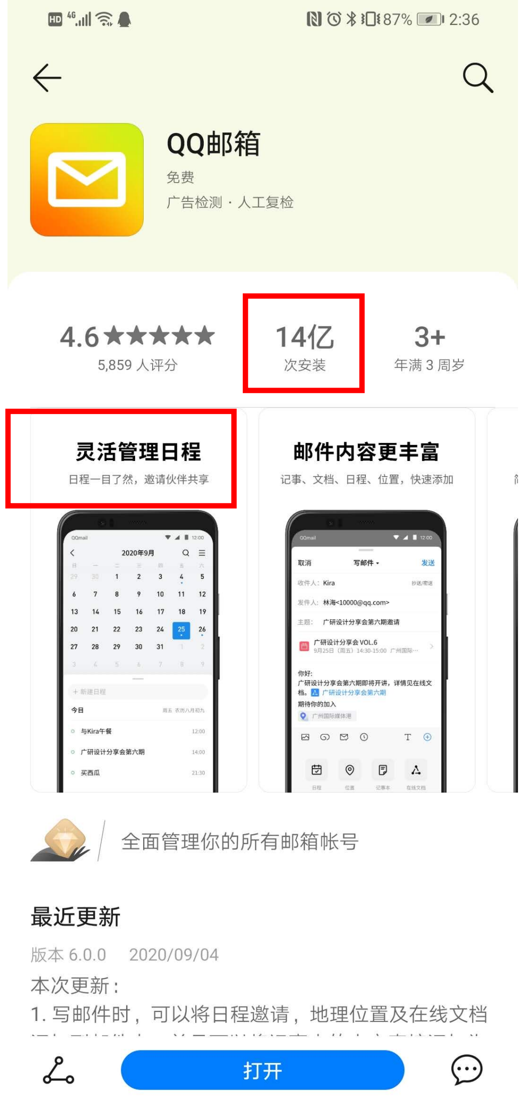
App ID	App name	#Installations	#Stars	Issue ID	Issue state	Cause setting	Consequence
1	APhotoManager	-	162	#175	Confirmed	Permission	Crash
2	A2DP Volume	100K-500K	71	#295	Fixed	Display	Crash
				#294	Fixed	Display	Data lost
				#291	Fixed	Display	Crash
				#290	Fixed	Display & Permission	Data lost
				#289	Confirmed	Developer	Crash
3	Always On	10M-50M	121	#2476	Confirmed	Language	Disrespect of Settings
				#2475	Confirmed	Language	Incomplete translation(5)
4	AnkiDroid	5M-10M	3.2K	#5407	Fixed	Permission	Stuck
5	AntennaPod	500K-1M	3.3K	#4227	Fixed	Network	Lack of refresh
6	Commons	50K-100K	649	#3906	Discussion	Location	Infinite loading
				#3134	Confirmed	Permission	Crash
7	ConnectBot	1M-5M	1.6K				
8	FillUp	100K-500K	29				
9	Forecastie	10K-50K	609	#505	Fixed	Permission	Lack of prompt
				#504	Fixed	Language	Incomplete translation(5)
				#358	Confirmed	Display	Data lost
10	Good Weather	5K-10K	196	#62	Waiting	Network	Infinite loading
				#61	Waiting	Location	Lack of prompt
				#55	Waiting	Language	Language confusion
11	Notepad	100K-500K	156				
12	Omni Notes	100K-500K	2.2K	#776	Fixed	Permission	Lack of prompt
				#775	Fixed	Location	Functionality failure
				#764	Fixed	Language	Disrespect of Settings
				#695	Fixed	Language	Incomplete translation(2)
13	Opensuduku	10K-50K	209	#93	Confirmed	Language	Incomplete translation(7)
14	RedReader	50K-100K	1.1K	#783	Discussion	Network	Infinite loading
				#749	Confirmed	Language	Incomplete translation(23)
15	Timber	100K-500K	6.4K	#459	Confirmed	Display	Data lost
				#458	Waiting	Permission	Crash
				#454	Waiting	Permission	Incomplete translation(9)
16	Vanilla Music	500K-1M	777	#1048	Waiting	Display	Crash
17	Wikipedia	50M-100M	1.3K				
18	OpenBikeSharing	1K-5K	58	#55	Confirmed	Display	Functionality failure
19	Suntimes	-	134	#420	Fixed	Location	Infinite loading
20	RadioBeacon	-	43	#249	Confirmed	Network	Stuck
				#234	Confirmed	Permission	Crash
21	RunnerUp	10K-50K	511	#923	Fixed	Permission	Lack of prompt
22	Amaze	1M-5M	3K	#1965	Fixed	Display & Permission	Black screen
				#1964	Fixed	Display & Permission	Data lost
				#1920	Fixed	Network	Lack of prompt
				#1919	Fixed	Display & Permission	Crash
				#1885	Fixed	Permission	Crash
23	Habits	1M-5M	3.6K	#620	Fixed	Display	Data lost
				#599	Fixed	Language	Incomplete translation(2)
24	Materialistic	100K-500K	2.1K	#1429	Waiting	Network	Lack of refresh

Results: Bug Finding in Industrial Apps



ID	App	Setting	Consequence
1	QQMail	Permission	Function failure
2	QQMail	Permission	Crash
3	Wechat	Permission	Function failure
4	Wechat	Permission	Function failure
5	Wechat	Language	Problematic UI display
6	Wechat	Language	Incomplete translation
7	Wechat	Network	Stuck
8	Wechat	Network	Function failure
9	CapCut	Network	Infinite loading
10	CapCut	Permission	Function failure
11	CapCut	Display&Permission	Problematic UI display
12	CapCut	Network	Function failure
13	TikTok	Network	Function failure
14	TikTok	Permission	Function failure
15	TikTok	Locaion	Function failure
16	AlipayHK	Language	Function failure
17	AlipayHK	Location	Function failure

A QQ-Mail Bug



HD 4G 87% 2:36

QQ邮箱

免费

广告检测 · 人工复检

4.6 ★★★★★
5,859 人评分

14亿
次安装

3+
年满 3 周岁

灵活管理日程
日程一目了然，邀请伙伴共享

邮件内容更丰富
记事、文档、日程、位置，快速添加

全面管理你的所有邮箱帐号

最近更新

版本 6.0.0　2020/09/04

本次更新：

1. 写邮件时，可以将日程邀请，地理位置及在线文档

- **Consequence:**

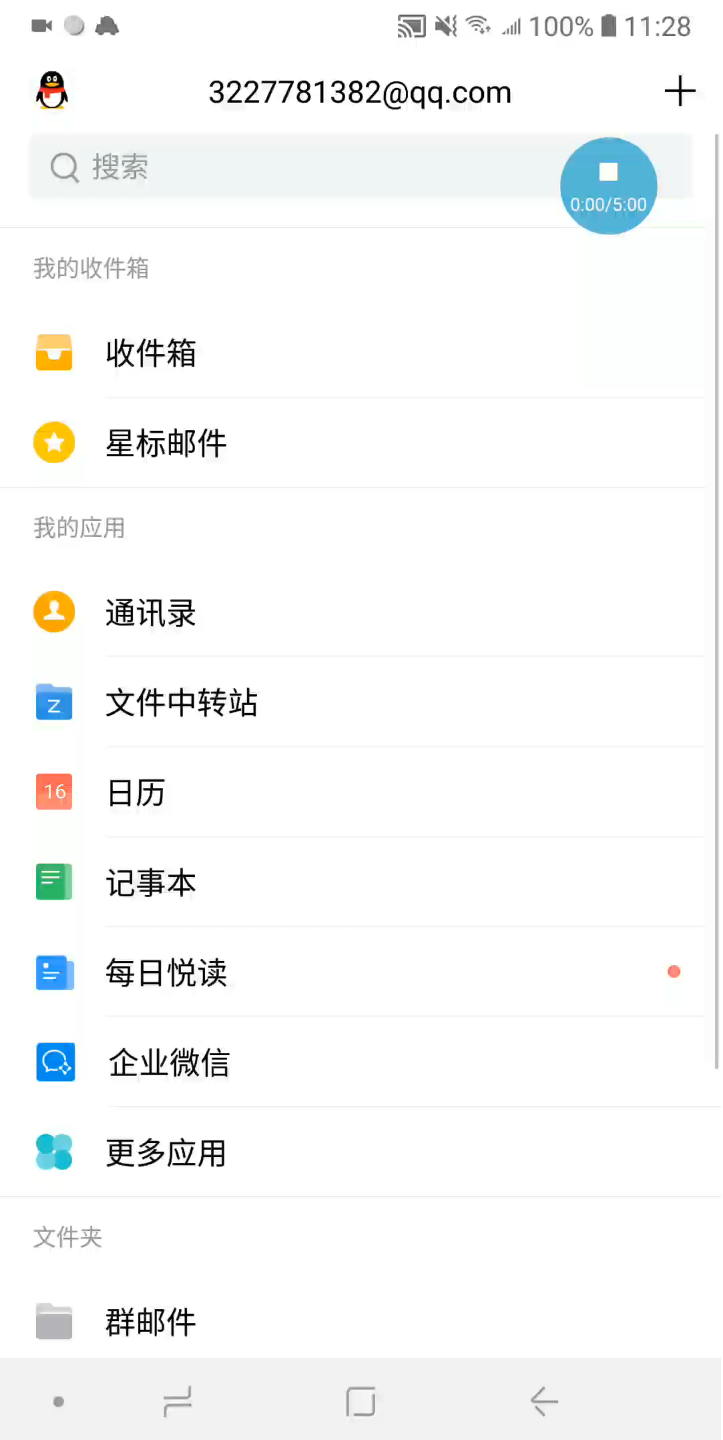
- cannot delete a to-do task

- **Root cause:**

- Fail to properly handle the calendar permission

- **Status:**

- Confirmed and fixed



100% 11:28

3227781382@qq.com

我的收件箱

收件箱

星标邮件

我的应用

通讯录

文件中转站

16 日历

记事本

每日悦读

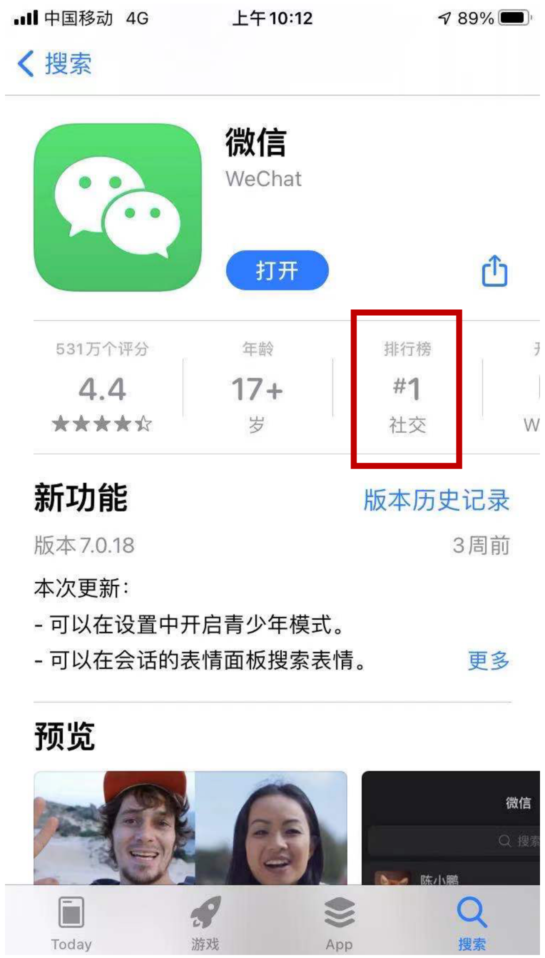
企业微信

更多应用

文件夹

群邮件

A WeChat Bug



中国移动 4G 上午10:12 89%

搜索

微信

WeChat

打开

531万个评分	年龄	排行榜
4.4	17+	#1
★★★★☆	岁	社交

新功能 版本历史记录

版本 7.0.18 3周前

本次更新：

- 可以在设置中开启青少年模式。
- 可以在会话的表情面板搜索表情。 更多

预览

- **Consequence:**

- User video/audio is leaked

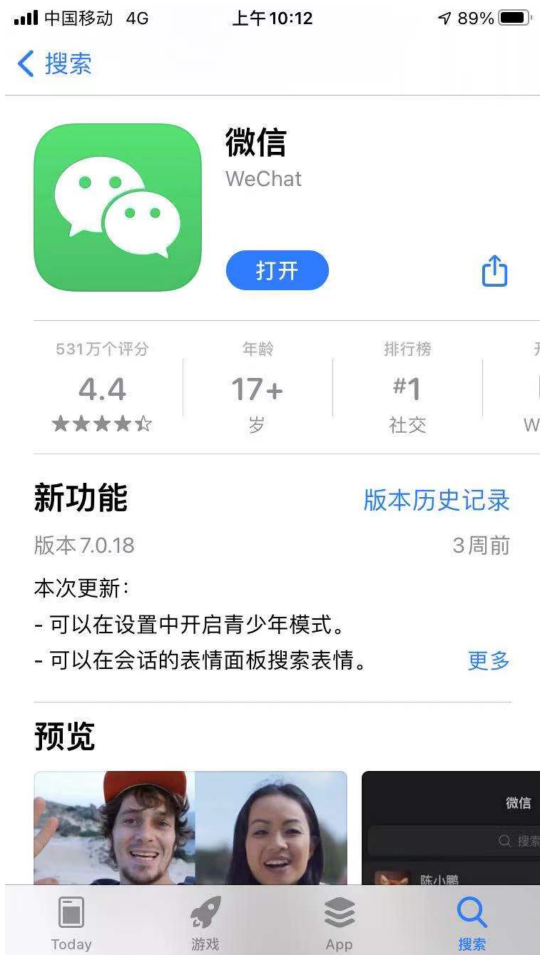
- **Root cause:**

- Fail to properly handle pop-up window permission
- Galaxy A6s, HuaWei nova 5 Pro

- **Status:**

- Confirmed and fixed

A WeChat Bug



微信

WeChat

531万个评分	年龄	排行榜
4.4	17+	#1
★★★★☆	岁	社交

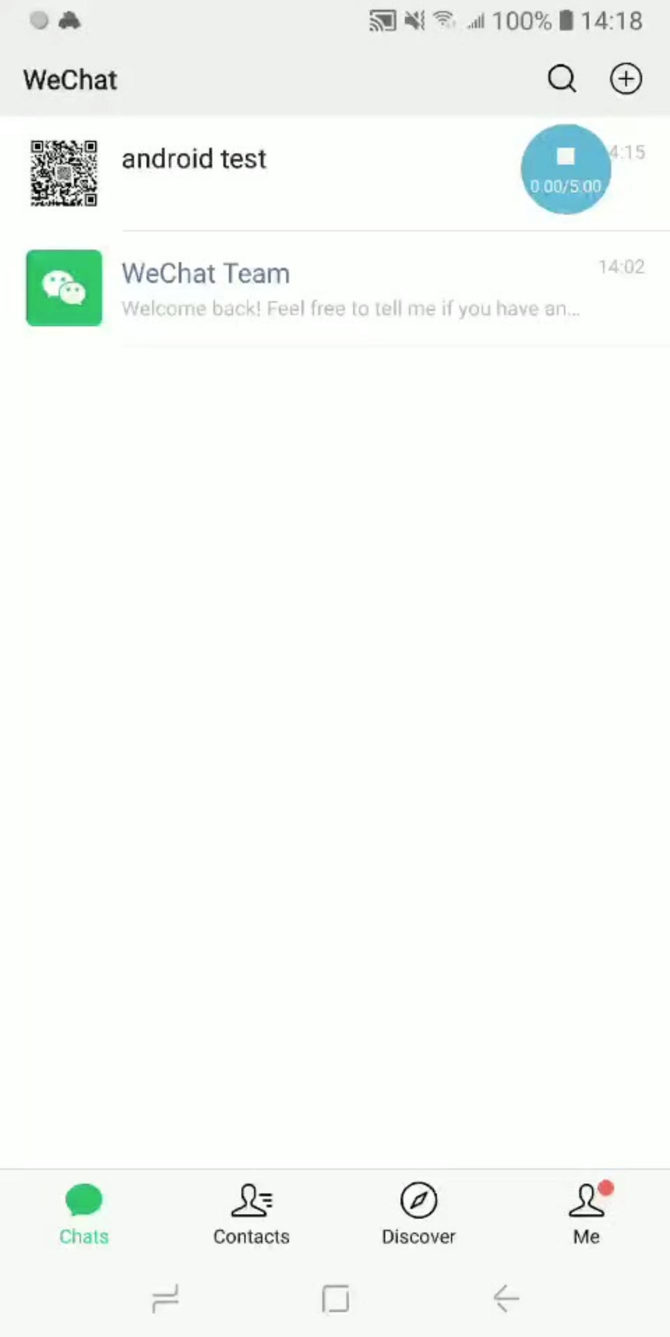
新功能 版本历史记录

版本7.0.18 3周前

本次更新：

- 可以在设置中开启青少年模式。
- 可以在会话的表情面板搜索表情。 更多

预览

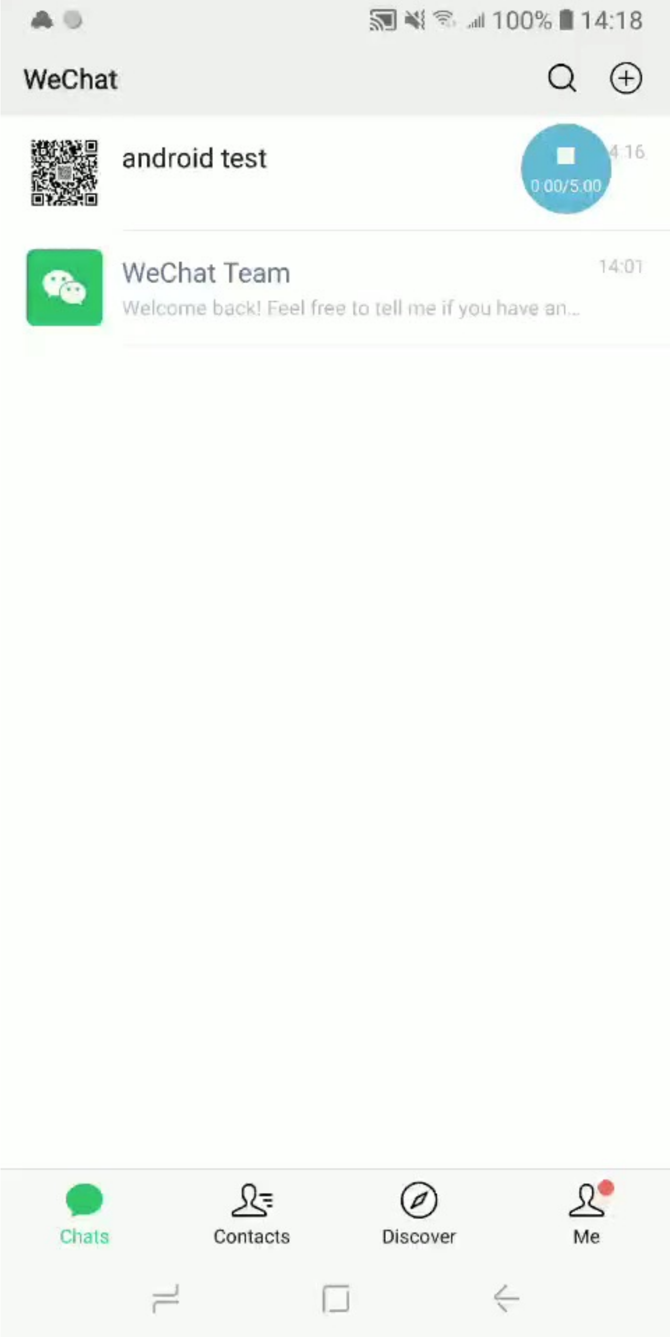


android test 4:15

WeChat Team 14:02

Welcome back! Feel free to tell me if you have an...

视频电话的 发起方



android test 4:16

WeChat Team 14:01

Welcome back! Feel free to tell me if you have an...

视频电话的 接收方

Lessons Learned

Metamorphic Testing (蜕变测试)

❑ An approach to both input generation (测试数据生成) and test result validation (测试结果验证)

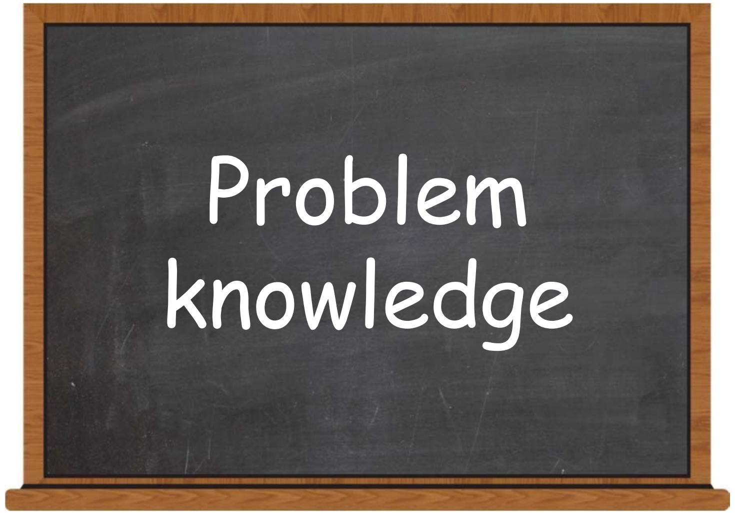
❑ A central element is a set of metamorphic relations (蜕变关系), which are *necessary* properties (必要性质) of the target function or algorithm *in relation to* multiple inputs and their expected outputs (多个输入和输出之间的关系)

Tsong Yueh Chen, Shing Chi Cheung, and Siu Ming Yiu. 1998. *Metamorphic testing: A new approach for generating next test cases*. Technical Report HKUST-CS98-01. Hong Kong University of Science and Technology, Hong Kong.

Lessons Learned

Lesson learned

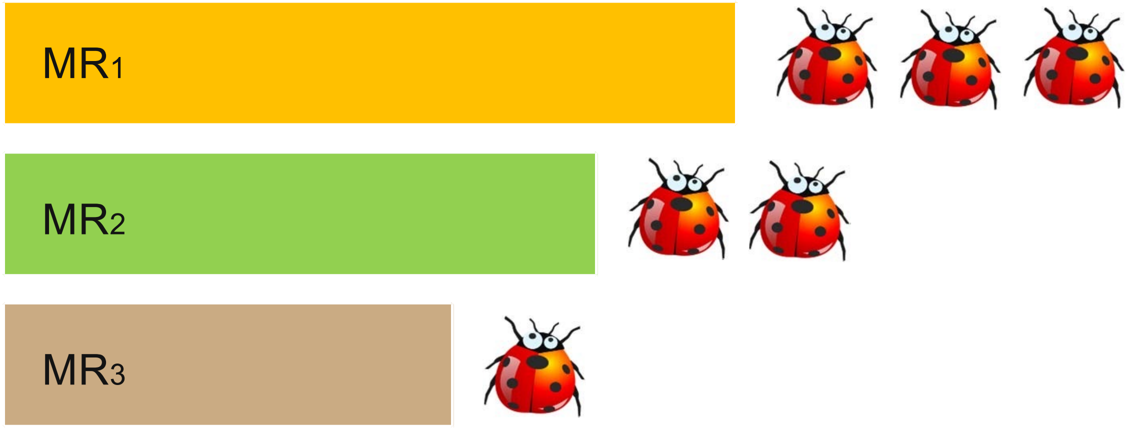
Metamorphic testing requires good knowledge of the problem domain.



Lessons Learned

Lesson learned

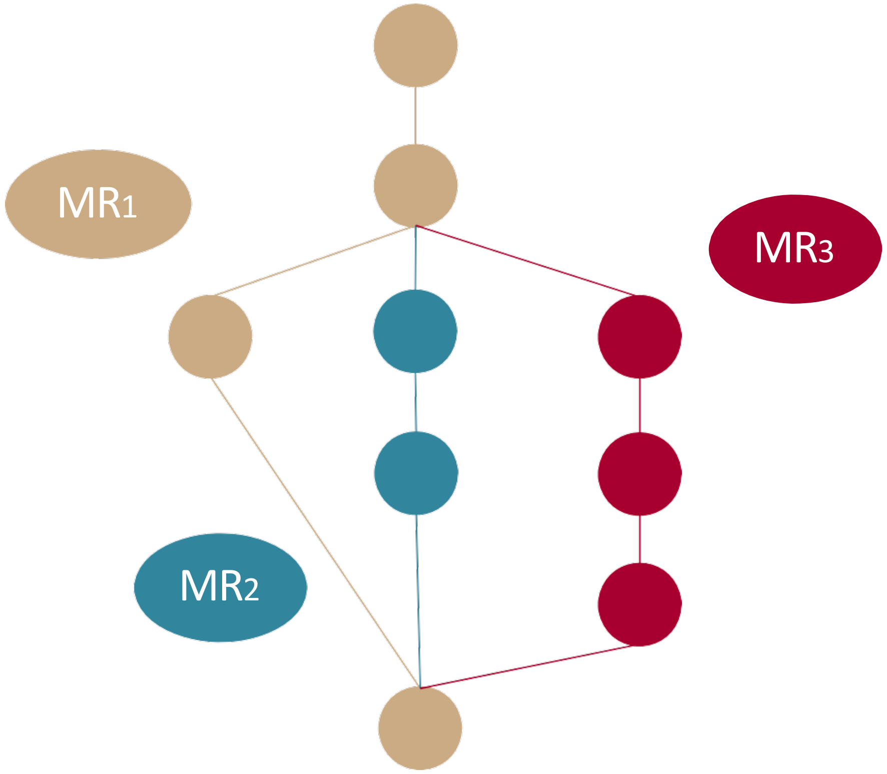
Different metamorphic relations can have different fault-detection capability.



Lessons Learned

Lesson learned

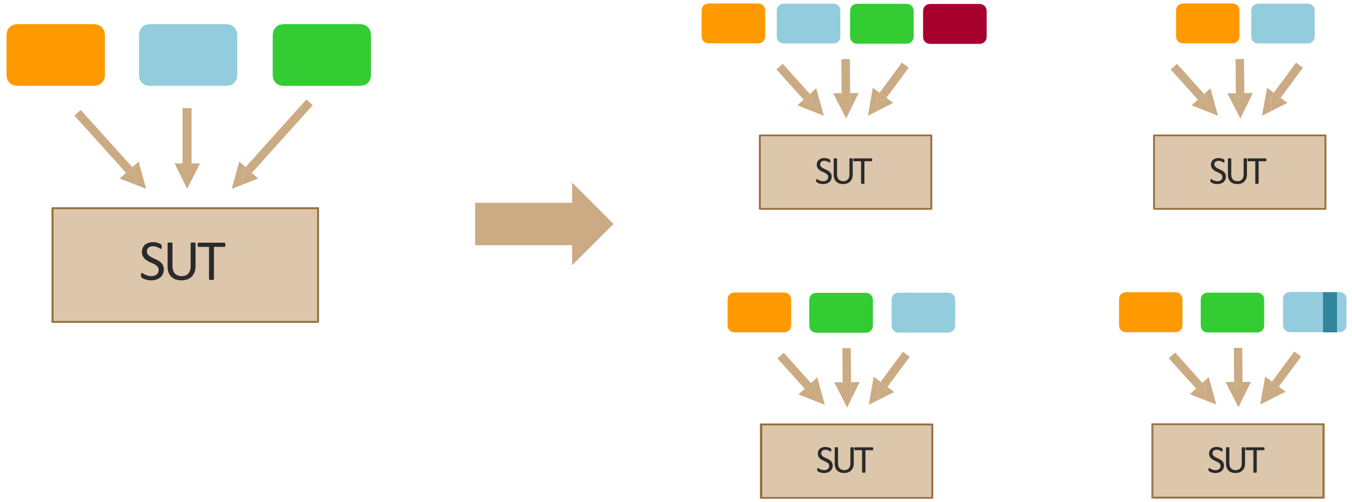
Metamorphic relations should be diverse so they exercise different parts of the program.



Lessons Learned

Lesson learned

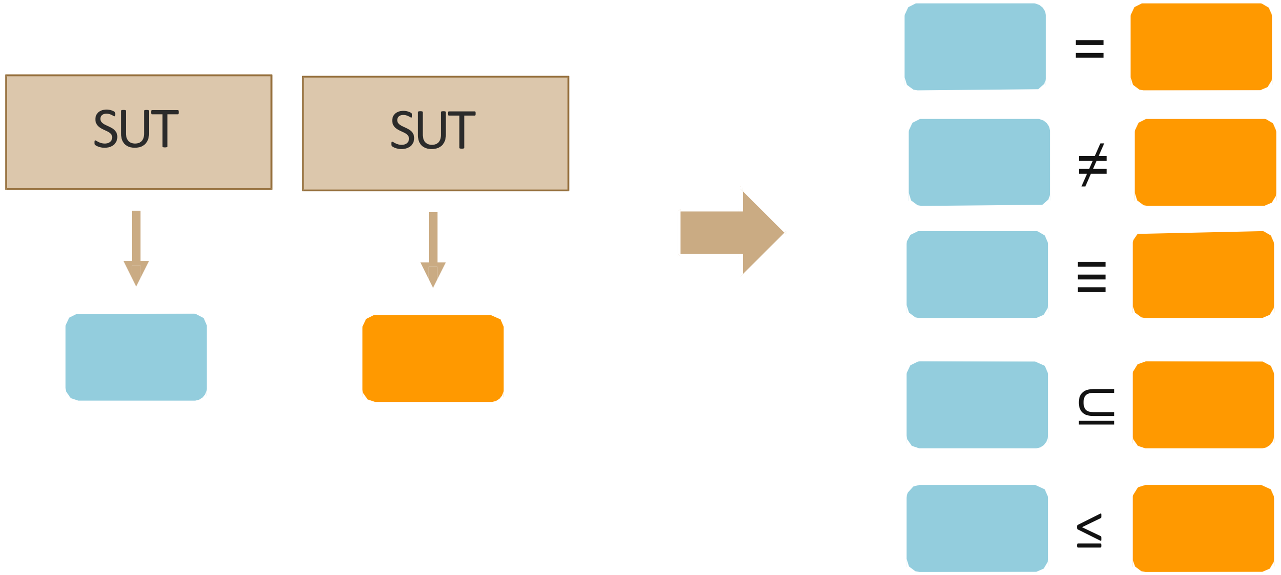
Two common approaches for the construction of metamorphic relations: **input-driven** vs output-driven



Lessons Learned

Lesson learned

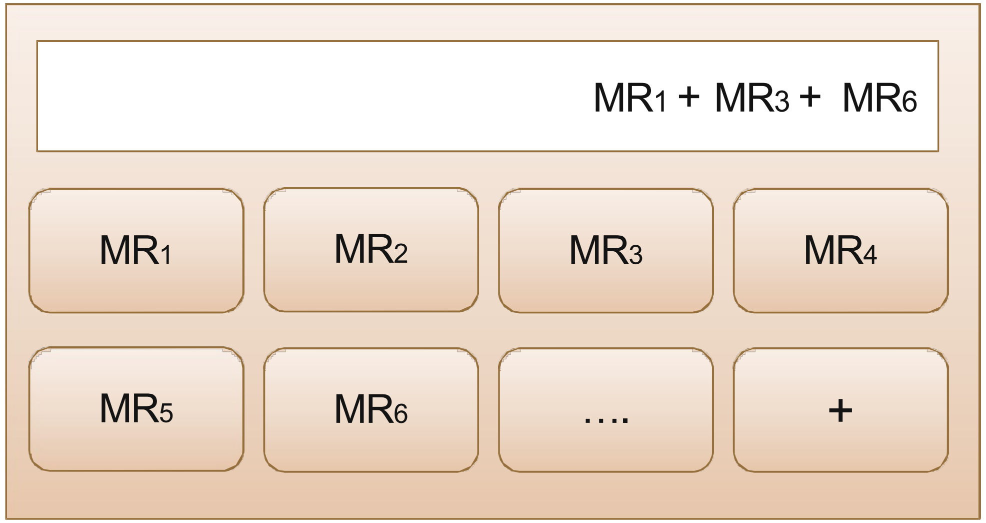
Two common approaches for the construction of metamorphic relations: input-driven vs **output-driven**



Lessons Learned

Lesson learned

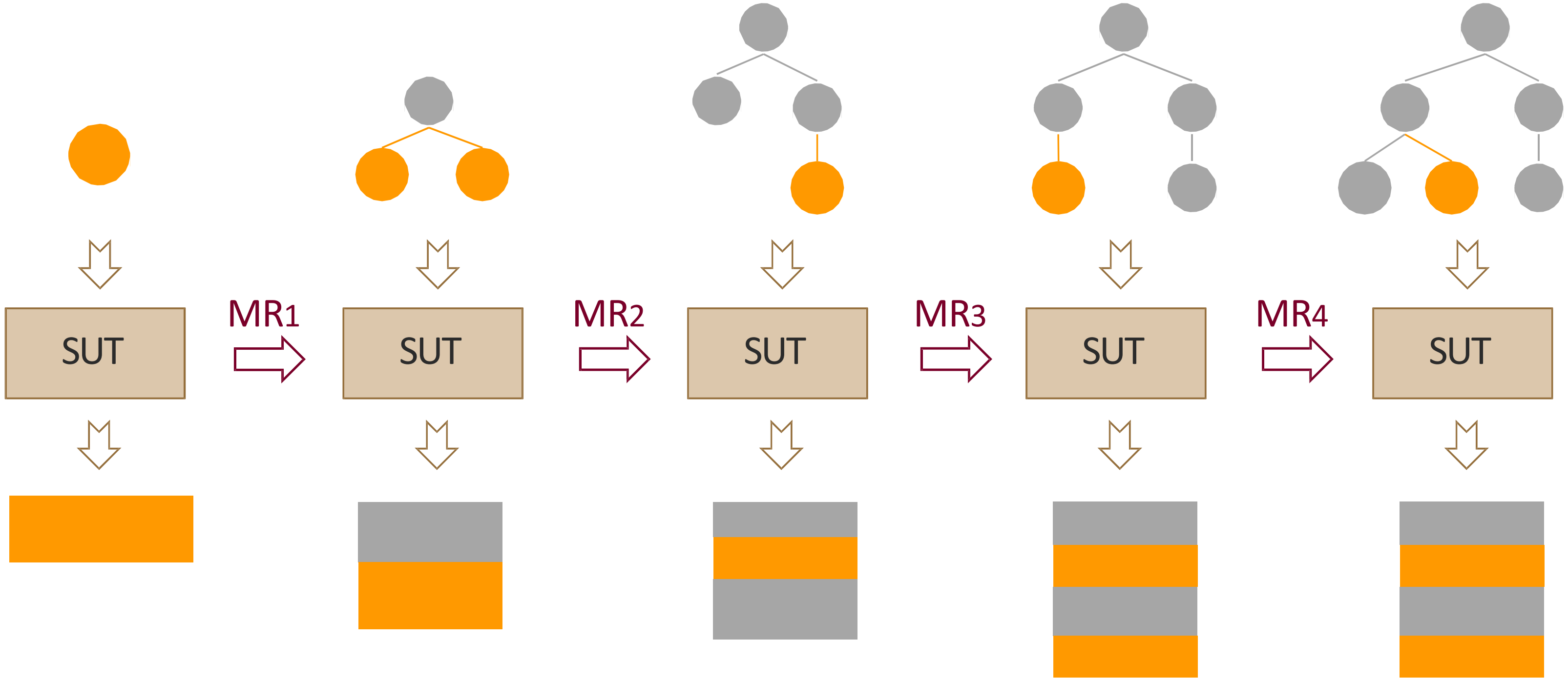
Metamorphic relations can be combined.



Lessons Learned

Lesson learned

Metamorphic relations can be combined.



Property-based Testing(基于性质的测试)

Revisit the Oracle Problem

A *sort* algorithm (which sorts a list of numbers)

```
void testSort()
{
   int [] list_a = [5, 4, 2, 7, 0, 1];
   int [] output = sort(list_a);
   int [] expected_output = [0, 1, 2, 4, 5, 7];
   assertEquals(output, expected_output);
}
```

Example-based Oracles (Assertions)

Revisit the Oracle Problem

A *sort* algorithm (which sorts a list of numbers)

```
void testSortMetamorphic(int [] list)
{
    int [] permutatedList = permutate(list);
    int [] output1 = sort(list);
    int [] output2 = sort(permutatedList);
    assertEquals(output1, output2);
}
```

Metamorphic Oracles

Are metamorphic oracles enough?

Revisit the Oracle Problem

A *sort* algorithm (which sorts a list of numbers)

```
@given(some.lists(some.integers()))
void testSortProperty(int [] list)
{
    int [] sorted_list = sort(list);
    for(int i=0; i<=len(sorted_list)-1; i++)
        assertTrue(sorted_list[i] <= sorted_list[i+1])
}
```

Revisit the Oracle Problem

A *sort* algorithm (which sorts a list of numbers)

```
@given(some.lists(some.integers()))
void testSortProperty(int [] list)
{
    int [] sorted_list = sort(list);
    for(int i=0; i<=len(sorted_list)-1; i++)
        assertTrue(sorted_list[i] <= sorted_list[i+1])
}
```

Property

Revisit the Oracle Problem

A *sort* algorithm (which sorts a list of numbers)

Random Input Generator

```
@given(some.lists(some.integers()))
void testSortProperty(int [] list)
{
    int [] sorted_list = sort(list);
    for(int i=0; i<=len(sorted_list)-1; i++)
        assertTrue(sorted_list[i] <= sorted_list[i+1])
}
```

Revisit the Oracle Problem

A *sort* algorithm (which sorts a list of numbers)

```
@given(some.lists(some.integers()))
void testSortProperty(int [] list)
{
    if (len(list) >= 100)
        return;
    int [] sorted_list = sort(list);
    for(int i=0; i<=len(sorted_list)-1; i++)
        assertTrue(sorted_list[i] <= sorted_list[i+1])
}
```

Conditions

Revisit the Oracle Problem

A *sort* algorithm (which sorts a list of numbers)

Random Input Generator

```
@given(some.lists(some.integers()))
void testSortProperty(int [] list)
{
    if (len(list) >= 100)
        return;
    int [] sorted_list = sort(list);
    for(int i=0; i<=len(sorted_list)-1; i++)
        assertTrue(sorted_list[i] <= sorted_list[i+1])
}
```

Conditions

Property => Assertions

Property-based Testing!

Property-based Testing (基于性质的测试)

- The term *property-based testing* originate from *"Property-Based Testing: A New Approach to Testing for Assurance"* by Fink and Bishop (SE Notes 1997)

- The approach was *popularized* by the work *"QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs"* by Claessen and Hughes (ICFP 2000)

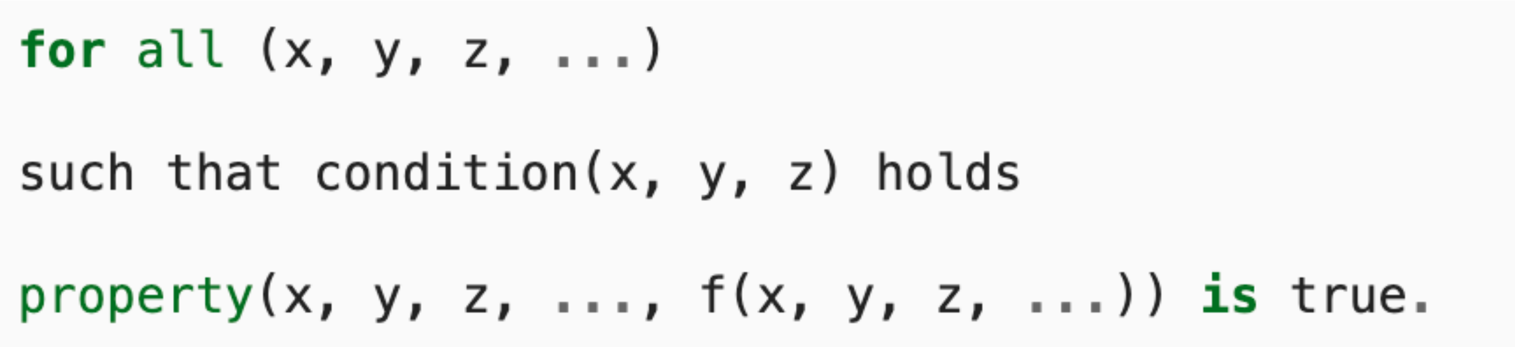
Property-based Testing (基于性质的测试)

- The term *property-based testing* originate from *"Property-Based Testing: A New Approach to Testing for Assurance"* by Fink and Bishop (SE Notes 1997)

- The approach was *popularized* by the work *"QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs"* by Claessen and Hughes (ICFP 2000)

- Property-based testing fundamentally depends on being able to write *abstract assertions* that are effectively *specifications*.

Property-based Testing (基于性质的测试)



```
for all (x, y, z, ...)

such that condition(x, y, z) holds

property(x, y, z, ..., f(x, y, z, ...)) is true.
```

where *x, y, z, ...* are input variables, and *f* is a function that we are testing. *condition* is a predicate that returns true if some property is true for *x, y, z, ...*, while *property* is a test oracle that returns *true* if the some property holds true between the inputs in *x, y, z, ...* and the output of the program *f(x, y, z, ...)*.

Property-based Testing Pattern

(如何产生有效的测试输入？)

1 Generating random inputs　**自动**生成随机输入

2 Ensuring the condition holds　**自动**检查前置条件是否满足

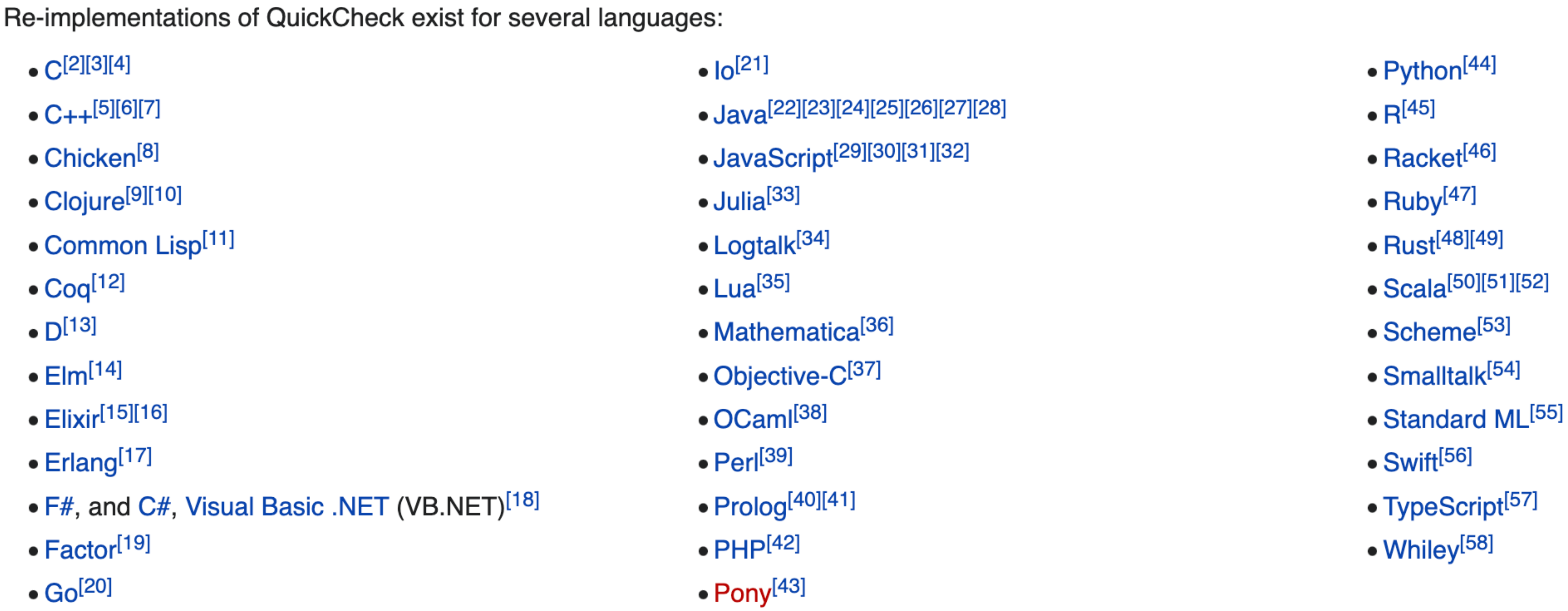
(性质从哪里来？怎么刻画和检查？)

3 Checking whether a property holds on those inputs　**自动**检查性质是否成立

4 Shrinking the counter-example input to obtain the minimal one　**自动**约减反例

Property-based Testing -- State-of-the-Art

Many Implementations for PBT



Re-implementations of QuickCheck exist for several languages:

- C[2][3][4]
- C++[5][6][7]
- Chicken[8]
- Clojure[9][10]
- Common Lisp[11]
- Coq[12]
- D[13]
- Elm[14]
- Elixir[15][16]
- Erlang[17]
- F#, and C#, Visual Basic .NET (VB.NET)[18]
- Factor[19]
- Go[20]
- Io[21]
- Java[22][23][24][25][26][27][28]
- JavaScript[29][30][31][32]
- Julia[33]
- Logtalk[34]
- Lua[35]
- Mathematica[36]
- Objective-C[37]
- OCaml[38]
- Perl[39]
- Prolog[40][41]
- PHP[42]
- Pony[43]
- Python[44]
- R[45]
- Racket[46]
- Ruby[47]
- Rust[48][49]
- Scala[50][51][52]
- Scheme[53]
- Smalltalk[54]
- Standard ML[55]
- Swift[56]
- TypeScript[57]
- Whiley[58]

https://en.wikipedia.org/wiki/QuickCheck

https://hypothesis.works/articles/quickcheck-in-every-language/

https://github.com/ksaaskil/introduction-to-property-based-testing

PBT frameworks

- QuickCheck
  (Haskell)
- junit-quickcheck
  (Java)
- Hypothesis
  (Python)
- FuzzTest

  (C/C++)

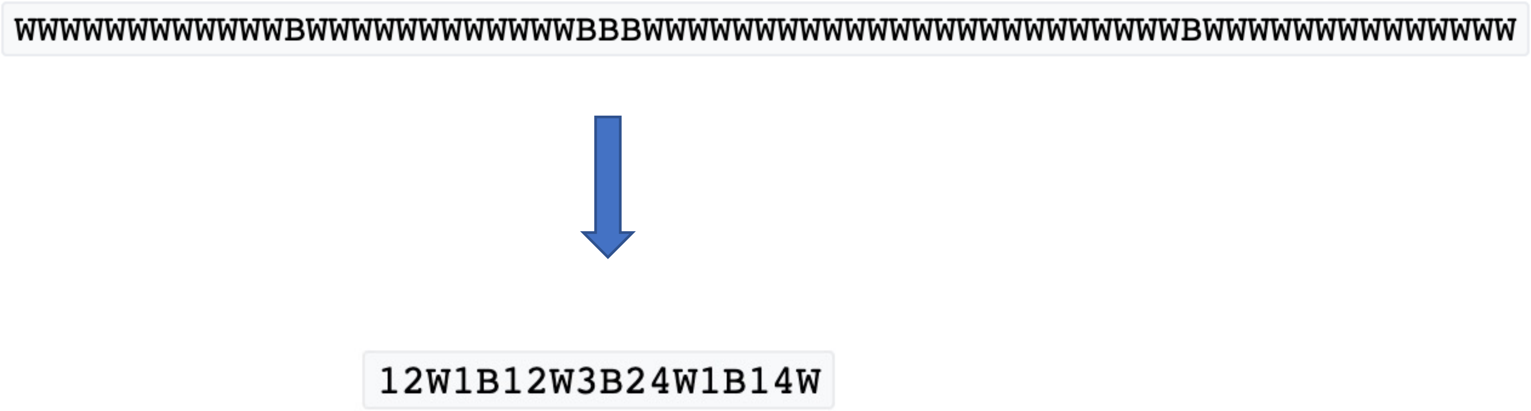
…

PBT frameworks --- Example

- QuickCheck
  (Haskell)
- junit-quickcheck
  (Java)
- Hypothesis
  (Python)
- FuzzTest

  (C/C++)

…



```
WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW
```

```
12W1B12W3B24W1B14W
```

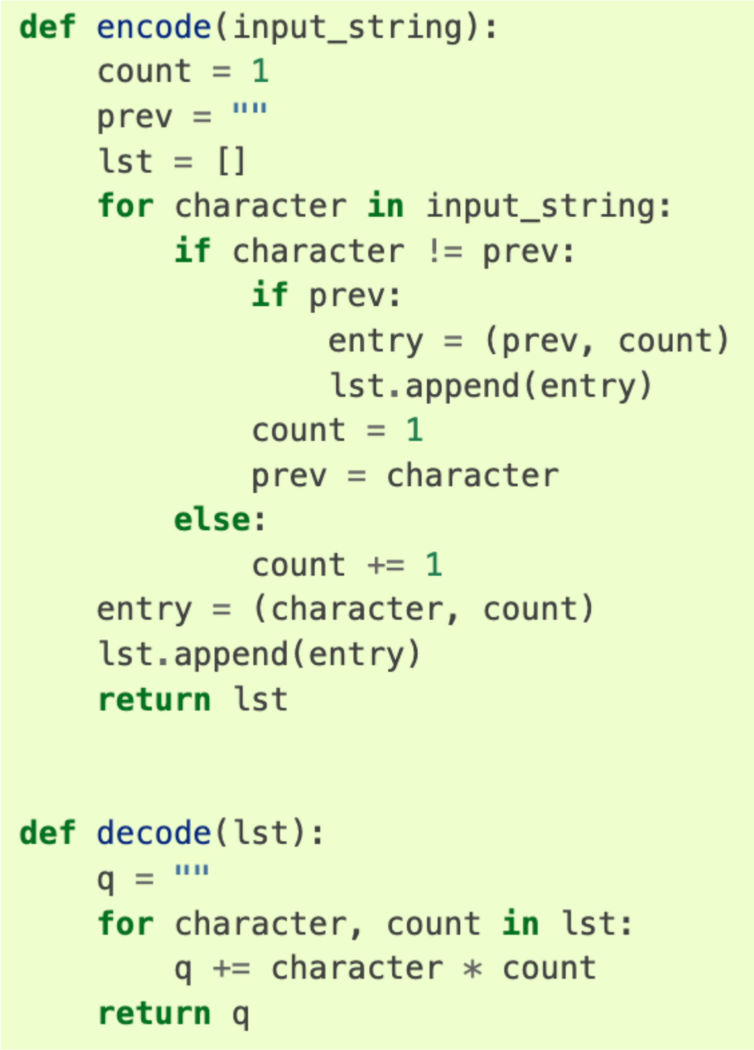
a run-length encoding system
https://en.wikipedia.org/wiki/Run-length_encoding

PBT frameworks --- Example

- QuickCheck
  (Haskell)
- junit-quickcheck
  (Java)
- Hypothesis
  (Python)
- FuzzTest

  (C/C++)

…



```python
def encode(input_string):
    count = 1
    prev = ""
    lst = []
    for character in input_string:
        if character != prev:
            if prev:
                entry = (prev, count)
                lst.append(entry)
            count = 1
            prev = character
        else:
            count += 1
    entry = (character, count)
    lst.append(entry)
    return lst


def decode(lst):
    q = ""
    for character, count in lst:
        q += character * count
    return q
```

a run-length encoding system

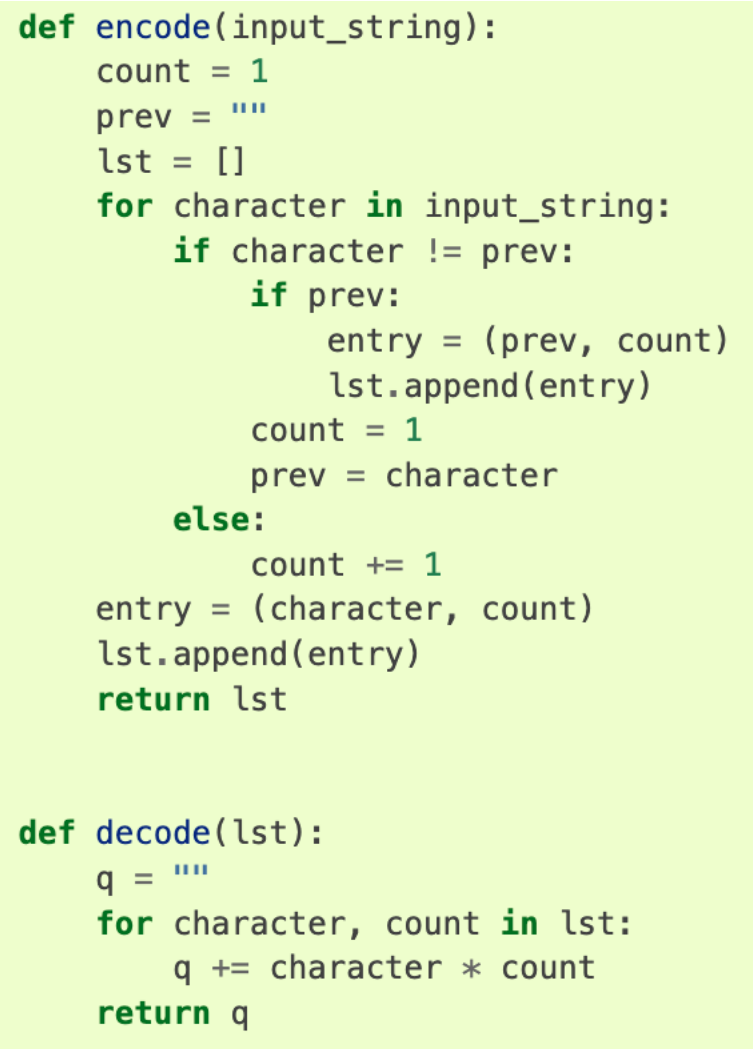
https://en.wikipedia.org/wiki/Run-length_encoding

PBT frameworks --- Example

- QuickCheck
  (Haskell)
- junit-quickcheck
  (Java)
- Hypothesis
  (Python)
- FuzzTest

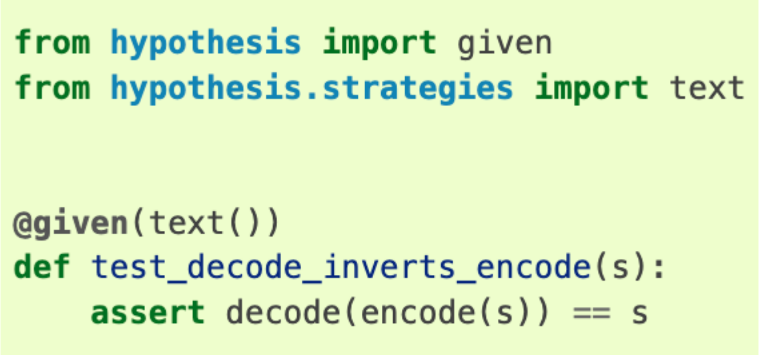
  (C/C++)

…



```python
def encode(input_string):
    count = 1
    prev = ""
    lst = []
    for character in input_string:
        if character != prev:
            if prev:
                entry = (prev, count)
                lst.append(entry)
            count = 1
            prev = character
        else:
            count += 1
    entry = (character, count)
    lst.append(entry)
    return lst


def decode(lst):
    q = ""
    for character, count in lst:
        q += character * count
    return q
```

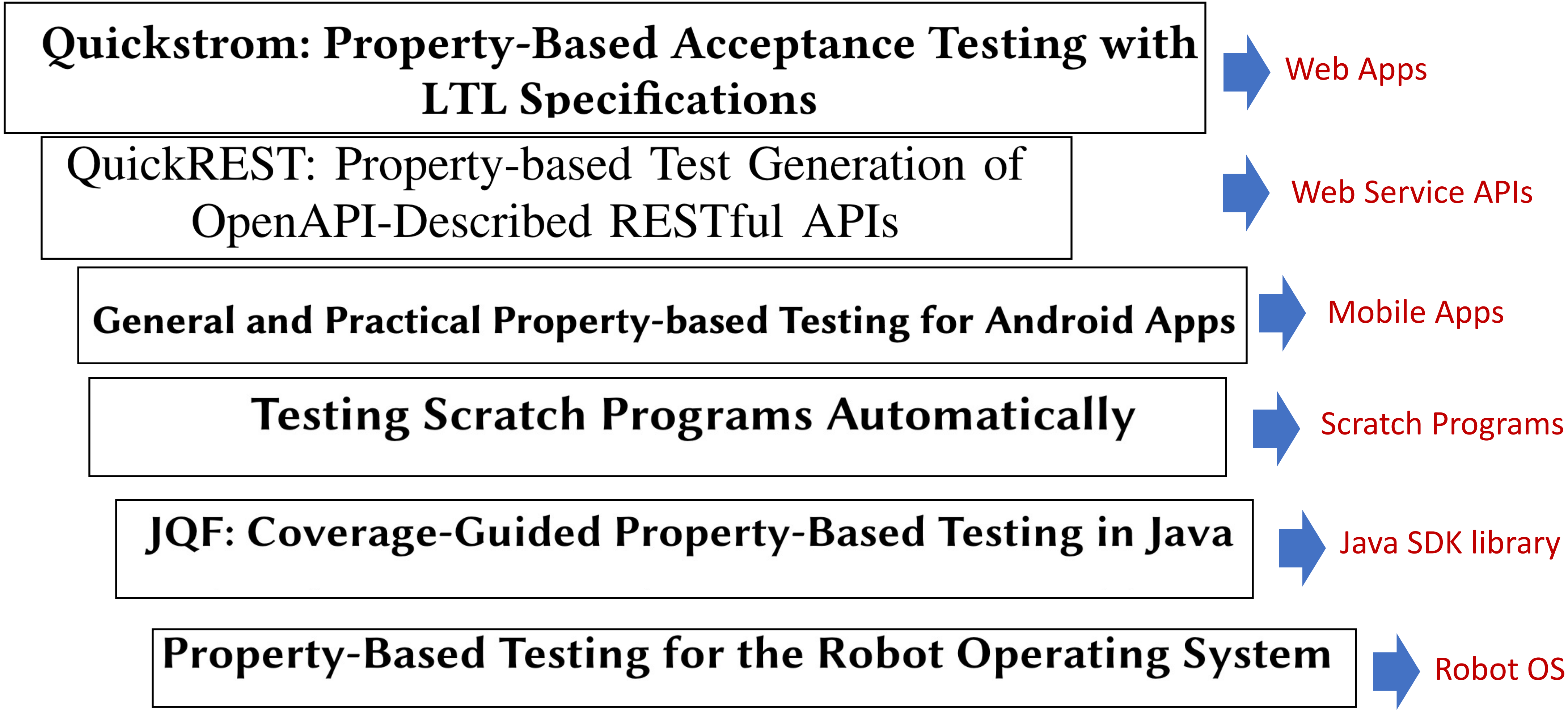
a run-length encoding system
https://en.wikipedia.org/wiki/Run-length_encoding



```python
from hypothesis import given
from hypothesis.strategies import text


@given(text())
def test_decode_inverts_encode(s):
    assert decode(encode(s)) == s
```

Property-based Tests

Successful Applications of PBT



Quickstrom: Property-Based Acceptance Testing with LTL Specifications → Web Apps

QuickREST: Property-based Test Generation of OpenAPI-Described RESTful APIs → Web Service APIs

General and Practical Property-based Testing for Android Apps → Mobile Apps

Testing Scratch Programs Automatically → Scratch Programs

JQF: Coverage-Guided Property-Based Testing in Java → Java SDK library

Property-Based Testing for the Robot Operating System → Robot OS

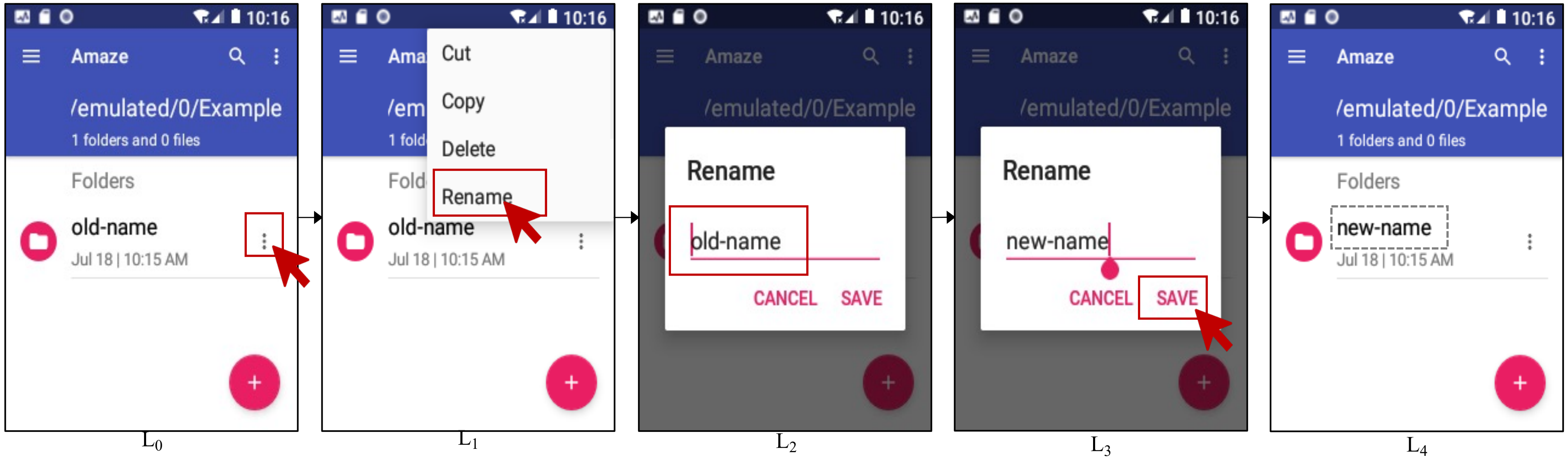
基于性质测试的探索案例

Data Manipulation Functionalities (DMF)

- **DMFs** are prevalent in mobile apps, which perform the CRUD operations (*create*, *read*, *update*, *delete*) to handle *app-specific data*.

Data Manipulation Functionalities (DMF)

- **DMFs** are prevalent in mobile apps, which perform the CRUD operations (*create*, *read*, *update*, *delete*) to handle *app-specific data*.



A “Rename” Function for app data “Folder”

Data Manipulation Functionalities (DMF)

- **DMFs** are prevalent in mobile apps, which perform the CRUD operations (*create*, *read*, *update*, *delete*) to handle *app-specific data*.

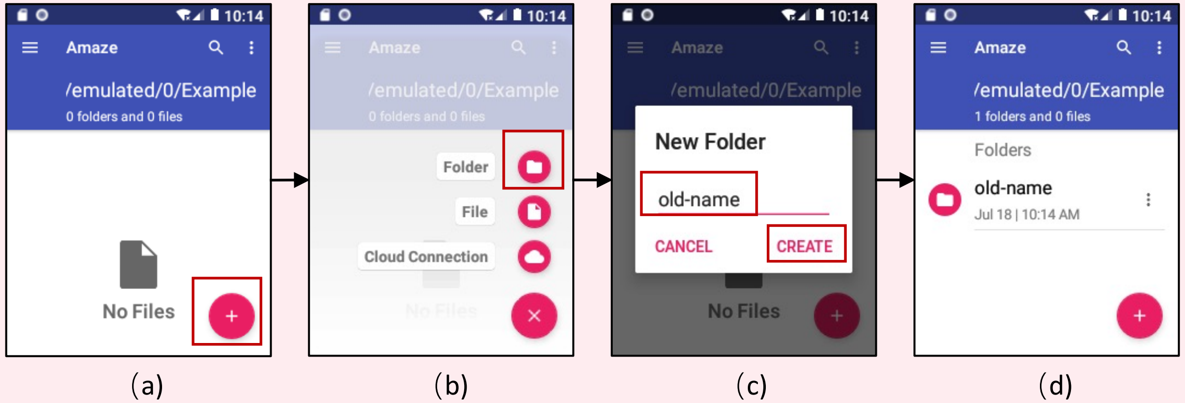
- Ensuring the **correctness** of these DMFs is *fundamentally important* for many core app functionalities.

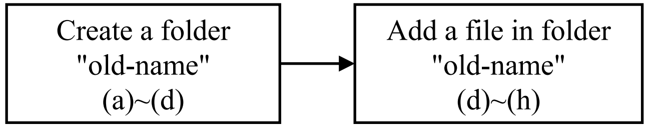
- The bugs related to DMFs are named as **data manipulation errors, DMEs**), are prevalent but difficult to find.

A Real Example of DME

Create a folder
"old-name"
(a)~(d)

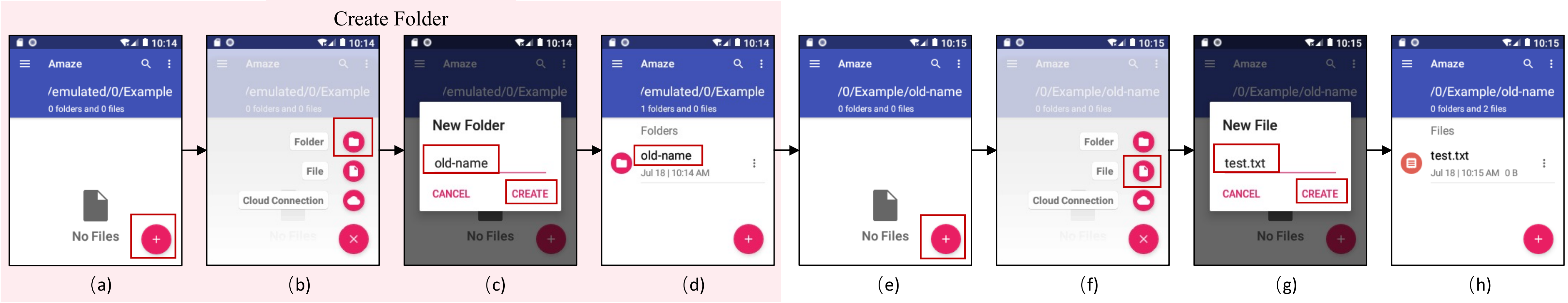
Create Folder

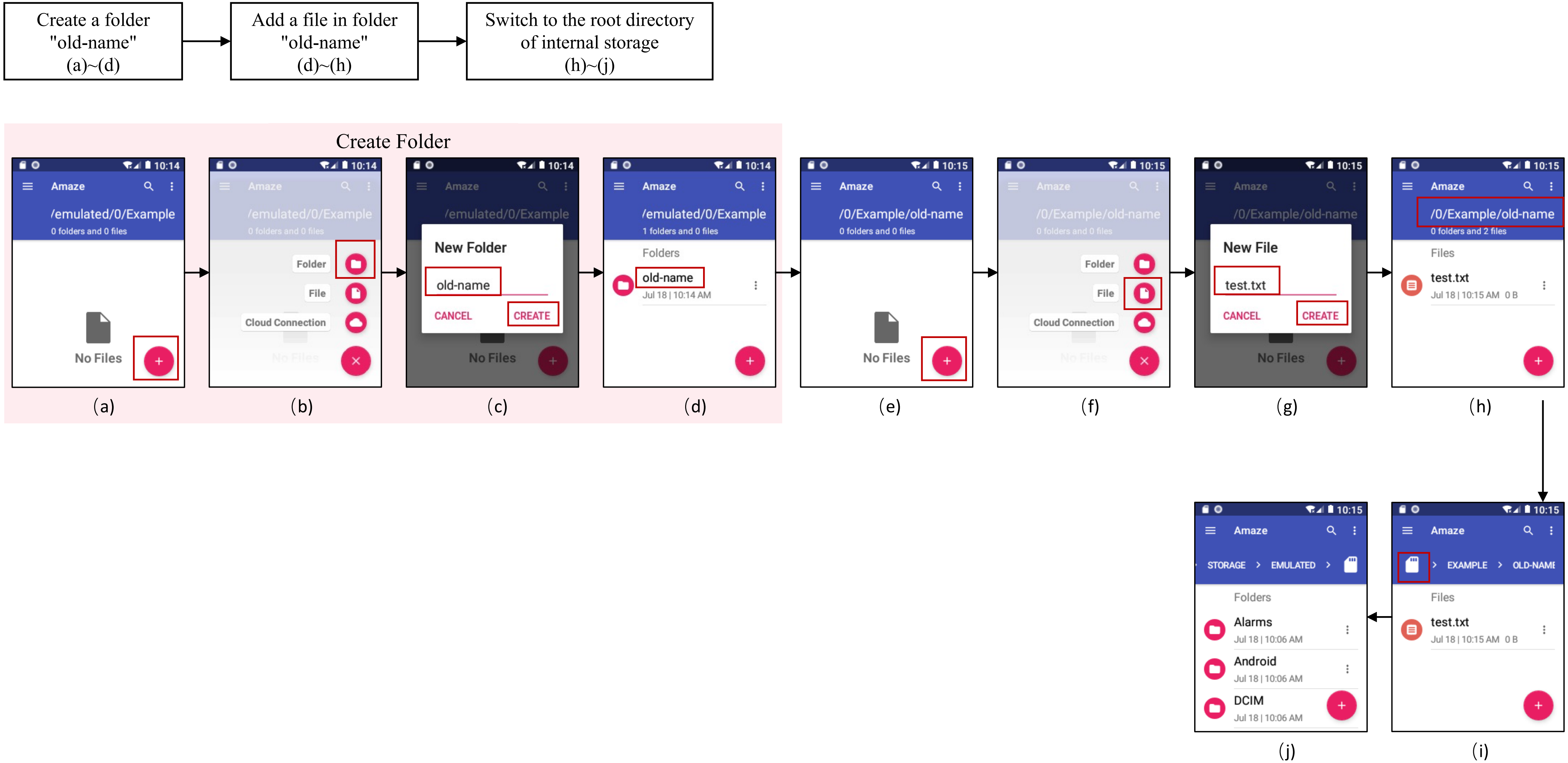




Create a folder "old-name" (a)~(d)

Add a file in folder "old-name" (d)~(h)



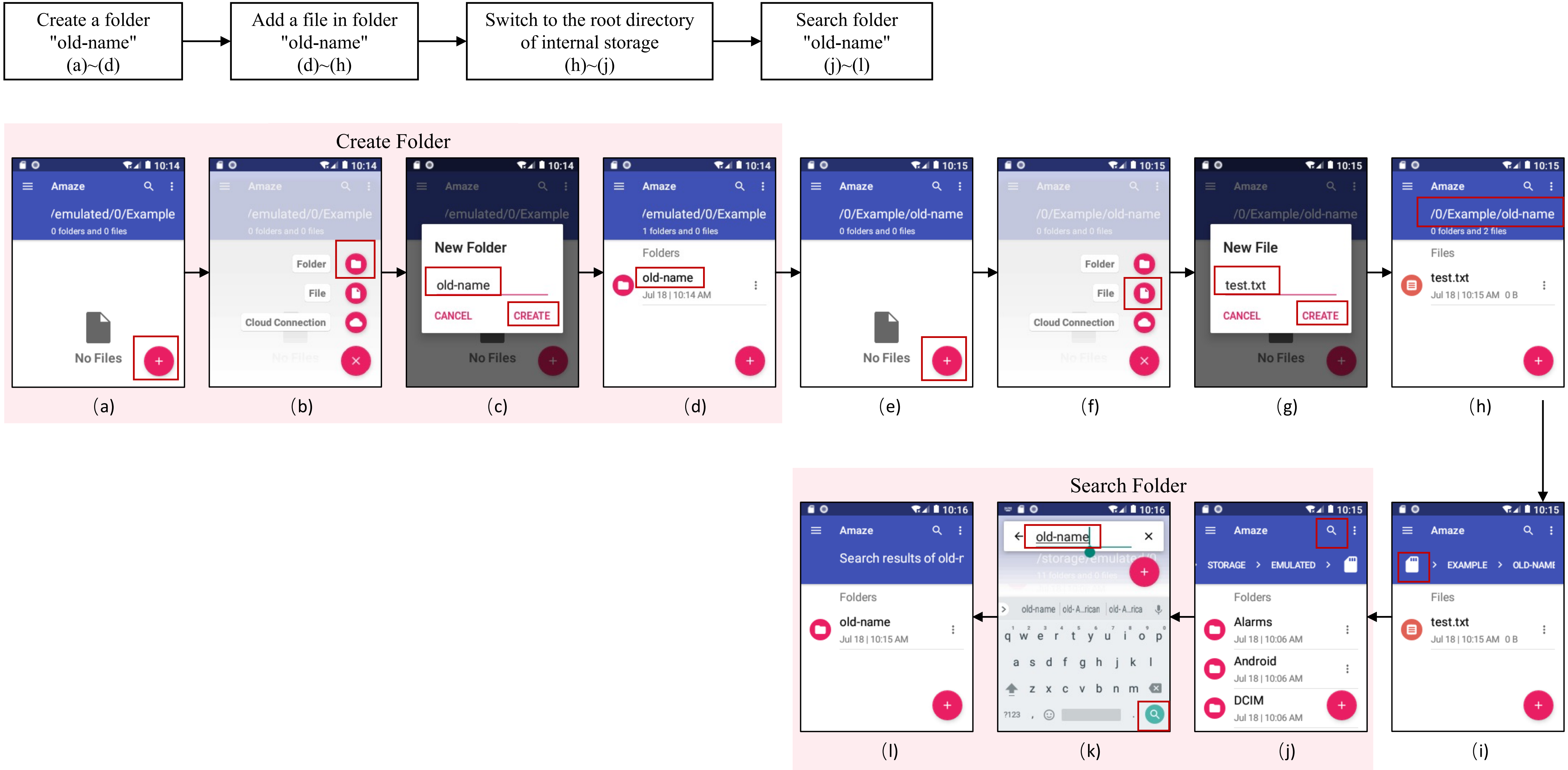


Create a folder "old-name" (a)~(d)

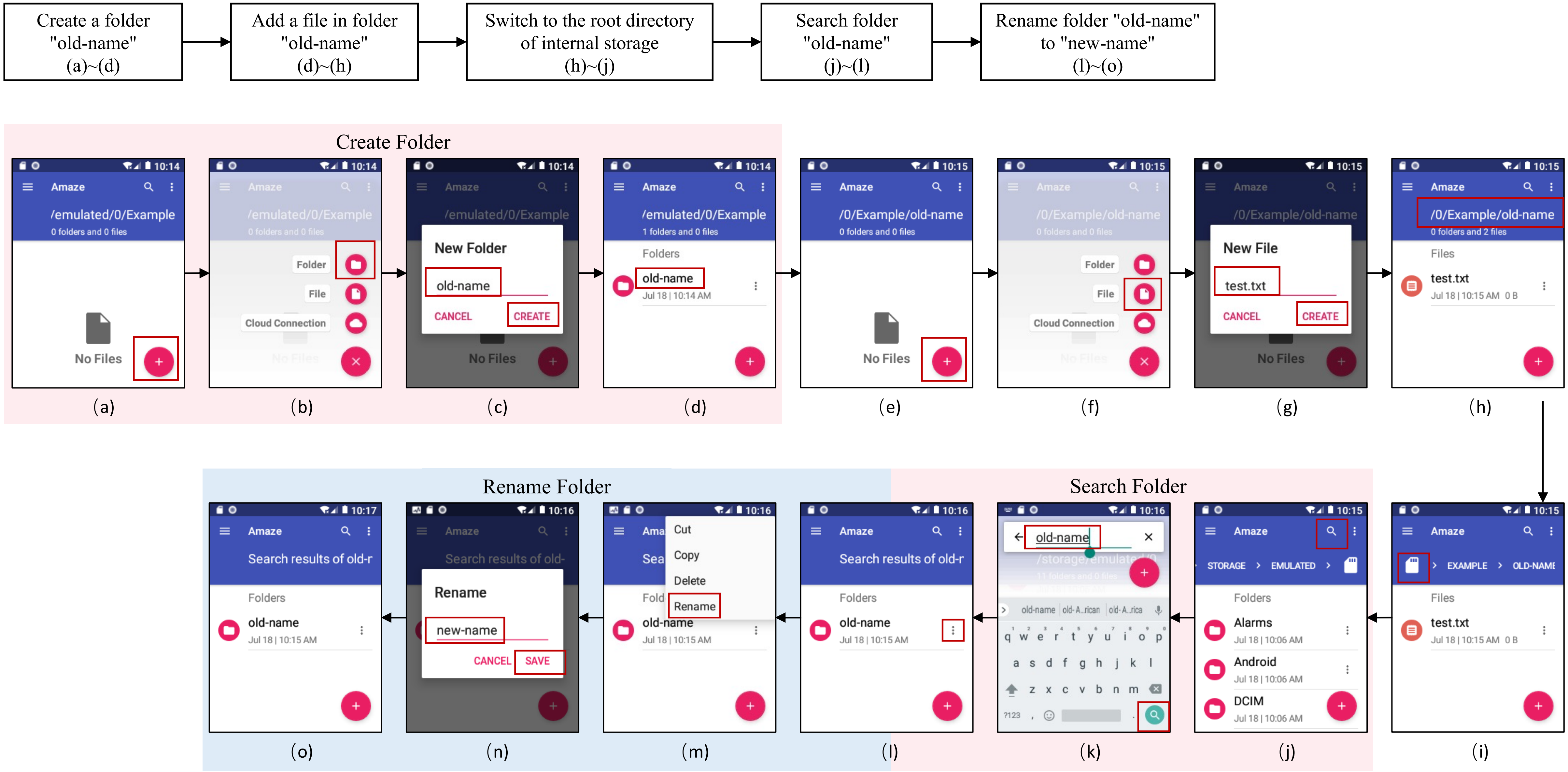
Add a file in folder "old-name" (d)~(h)

Switch to the root directory of internal storage (h)~(j)

Create Folder



Create a folder "old-name" (a)~(d) → Add a file in folder "old-name" (d)~(h) → Switch to the root directory of internal storage (h)~(j) → Search folder "old-name" (j)~(l)



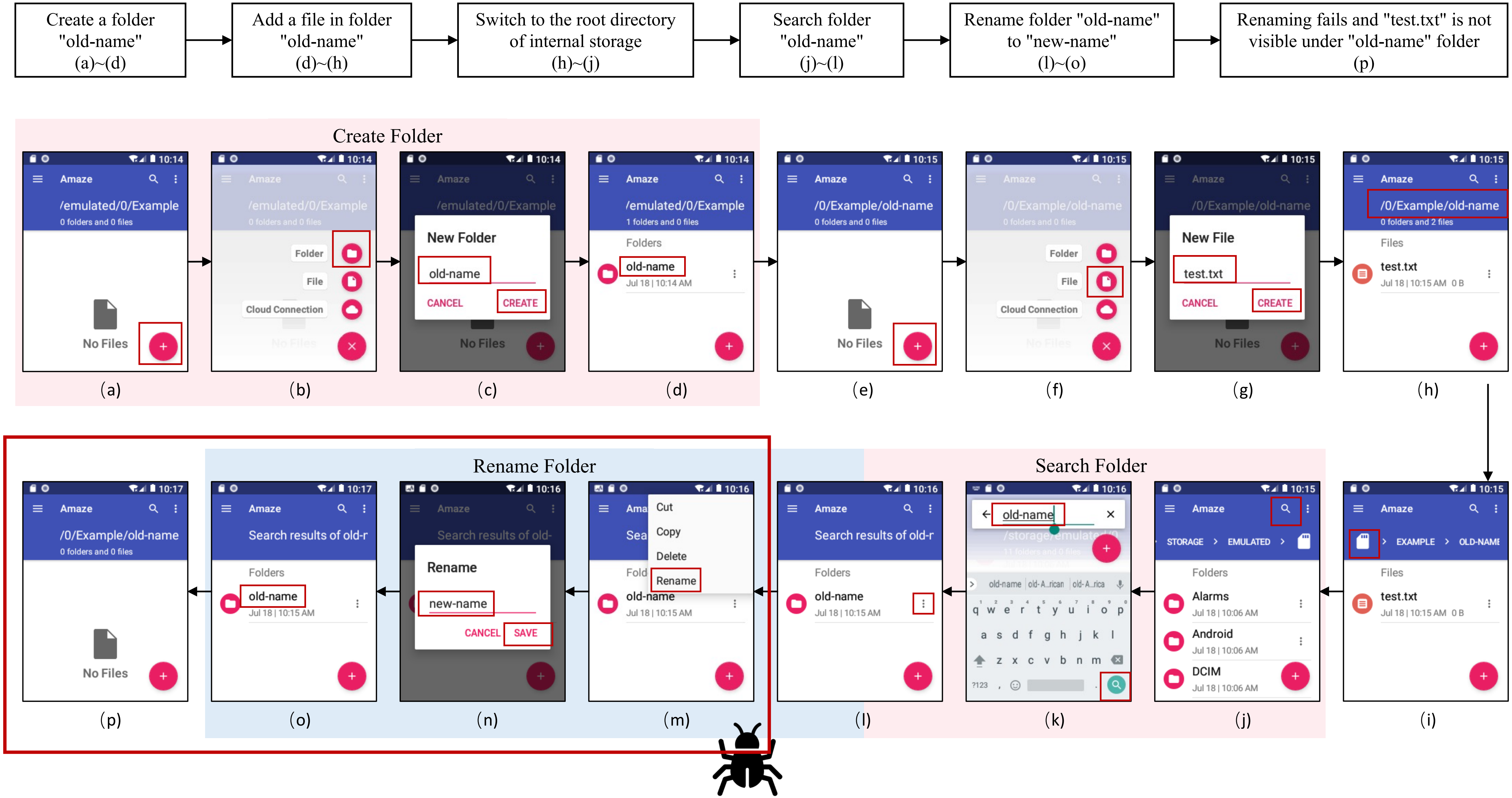
Create a folder "old-name" (a)~(d)

Add a file in folder "old-name" (d)~(h)

Switch to the root directory of internal storage (h)~(j)

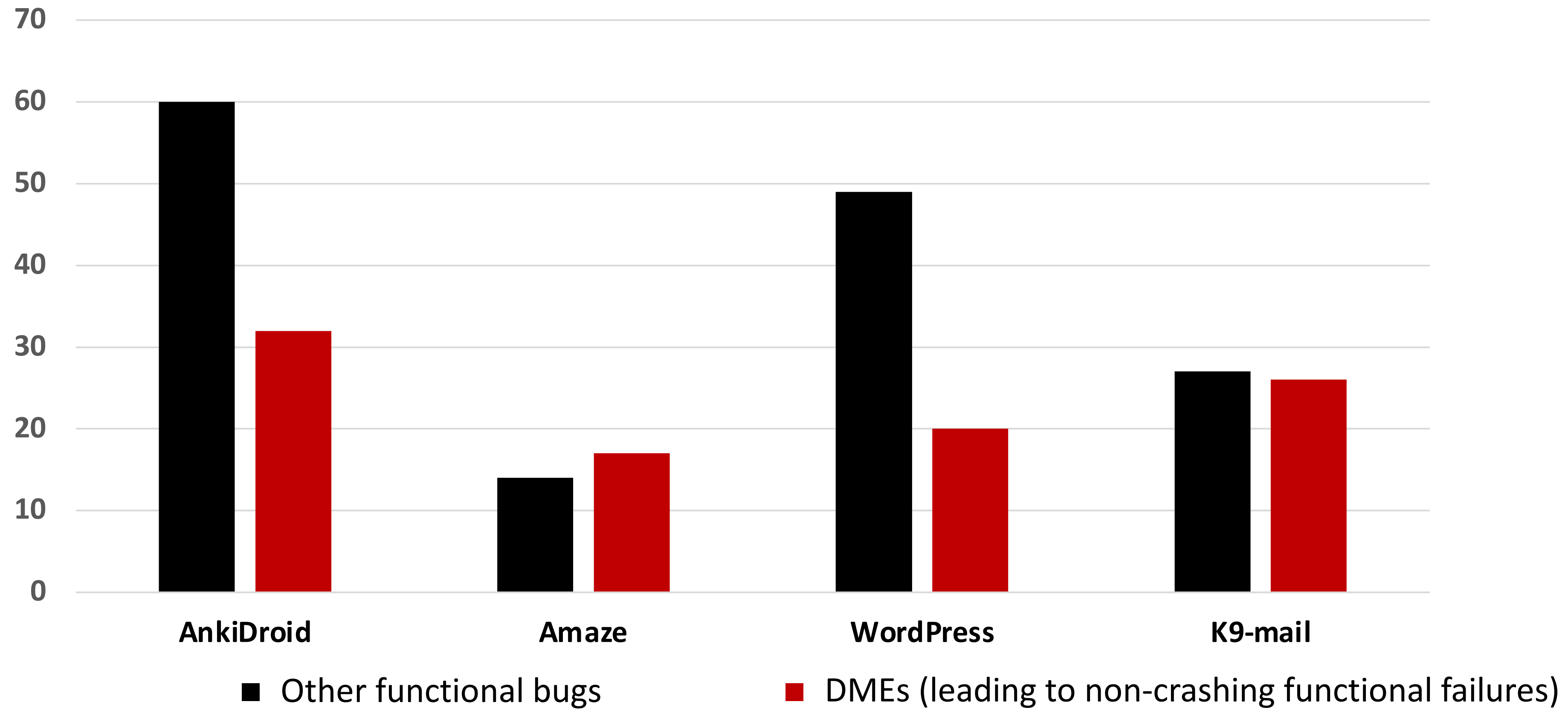
Search folder "old-name" (j)~(l)

Rename folder "old-name" to "new-name" (l)~(o)



Create a folder "old-name" (a)~(d) → Add a file in folder "old-name" (d)~(h) → Switch to the root directory of internal storage (h)~(j) → Search folder "old-name" (j)~(l) → Rename folder "old-name" to "new-name" (l)~(o) → Renaming fails and "test.txt" is not visible under "old-name" folder (p)

Prevalence of DMEs



*Bugs are collected from the issue repositories of these apps on GitHub from August 2018 and July 2021.

How to use *property-based testing* to find such data manipulation errors?

How to *Specify* the Properties of DMFs?

How to *Specify* the Properties of DMFs?

An app *property* φ of some functionality F can be specified as:

$$\phi = \langle \text{Pre}, \text{E}, \text{Post} \rangle$$

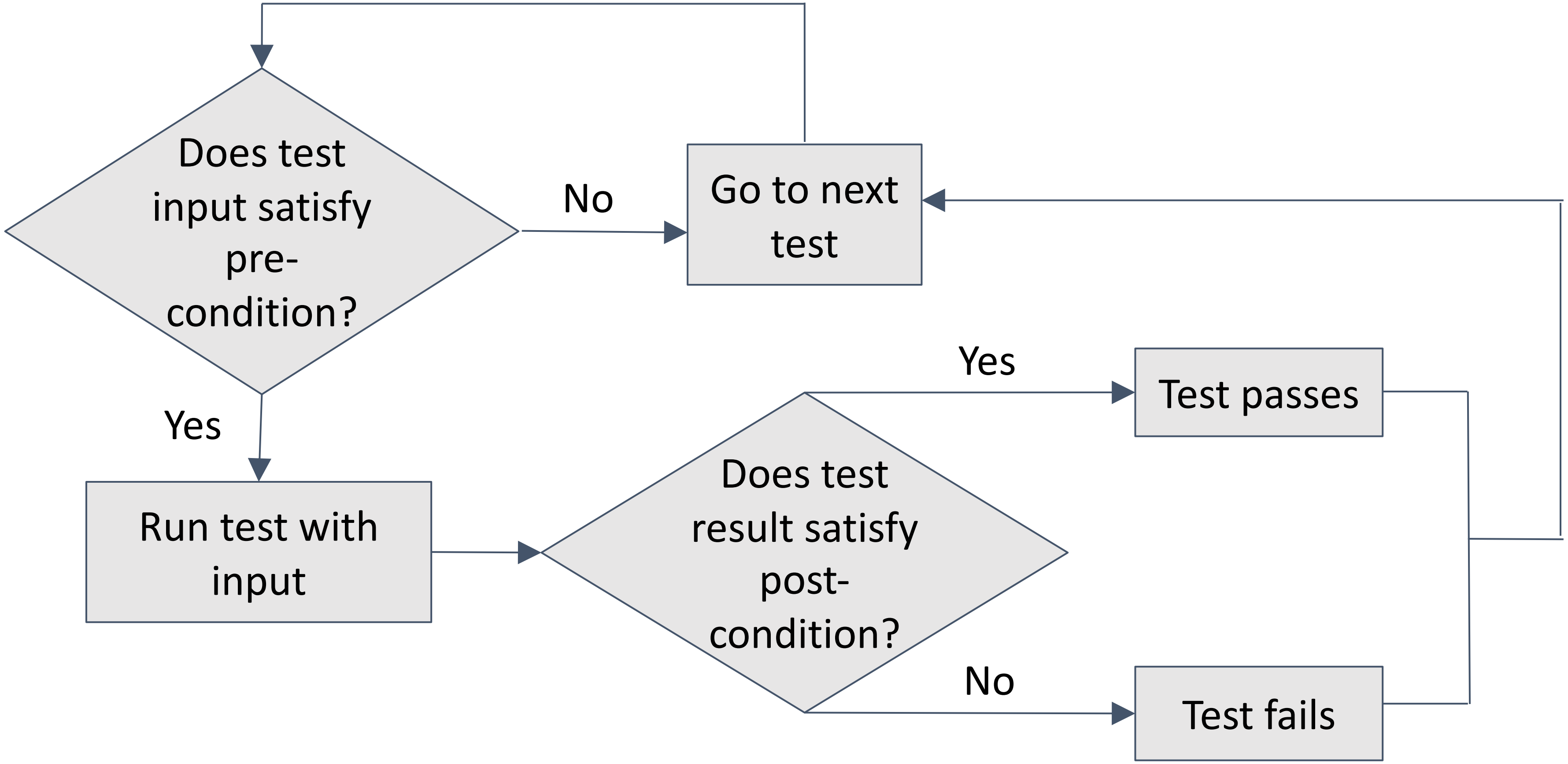
- ❖ E denotes F in the form of an event trace
- ❖ Pre is the *precondition* that must hold before execution of E
- ❖ Post is the *postcondition* defining the effect on the app state after executing E

How to *Check* the Properties of DMFs?

When an app *property* φ of some functionality F is specified as:

$$\phi = \langle \text{Pre, E, Post} \rangle$$

Using Pre- and Post-Conditions


Does test input satisfy pre-condition?
No
Go to next test
Yes
Run test with input
Does test result satisfy post-condition?
Yes
Test passes
No
Test fails

How to *Check* the Properties of DMFs?

When an app *property* φ of some functionality F is specified as:

$$\phi = \langle \text{Pre, E, Post} \rangle$$

How to *Check* the Properties of DMFs?

When an app *property* φ of some functionality F is specified as:

$$\phi = \langle \text{Pre, E, Post} \rangle$$

- ❖Generate many random inputs (*random UI event traces in our context*) to check ϕ. If some input satisfying Pre but fasifying Post after executing E, a DME is found.

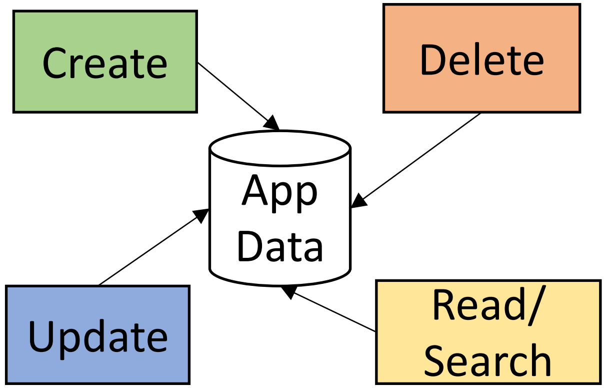
Let s be some app state, s ⊨ Pre and s'= E(s), if s'⊭ Post, then ϕ is violated.

How to Effectively *Check* the Properties of DMFs?

- Randomly ***interleave the relevant DMFs*** on the shared app data *(with other possible random events)* to exhibit *diverse* app states.

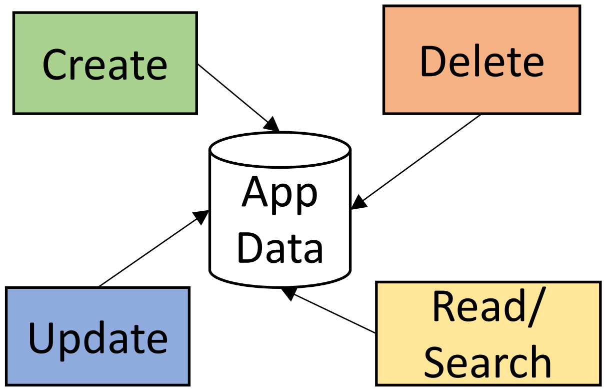
How to Effectively *Check* the Properties of DMFs?

- Randomly ***interleave the relevant DMFs*** on the shared app data *(with other possible random events)* to exhibit *diverse* app states.



How to Effectively *Check* the Properties of DMFs?

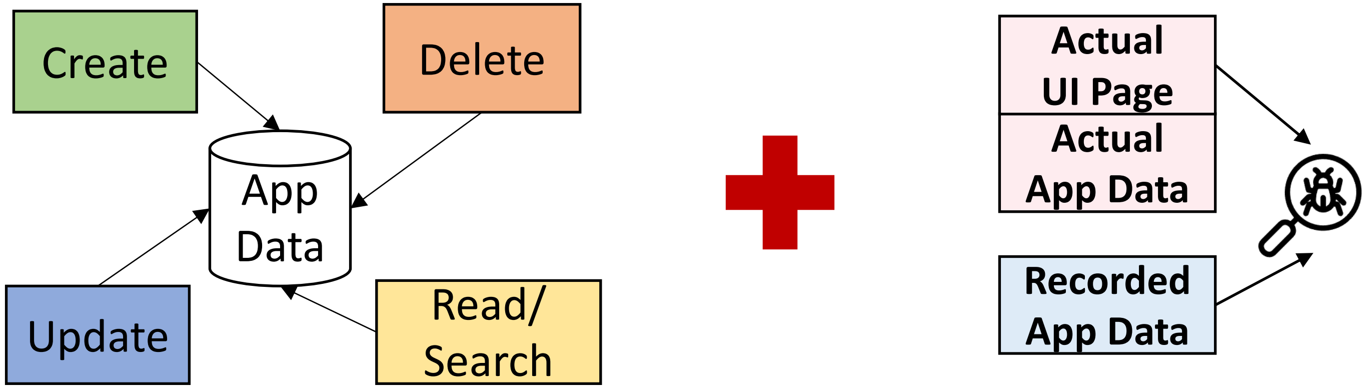
- Randomly ***interleave the relevant DMFs*** on the shared app data *(with other possible random events)* to exhibit *diverse* app states.



- Use ***an abstract data model*** to simulate the data update effect of E, thus facilitating property checking; and use the consistency between the app data and UI layouts to check the property.

How to Effectively *Check* the Properties of DMFs?

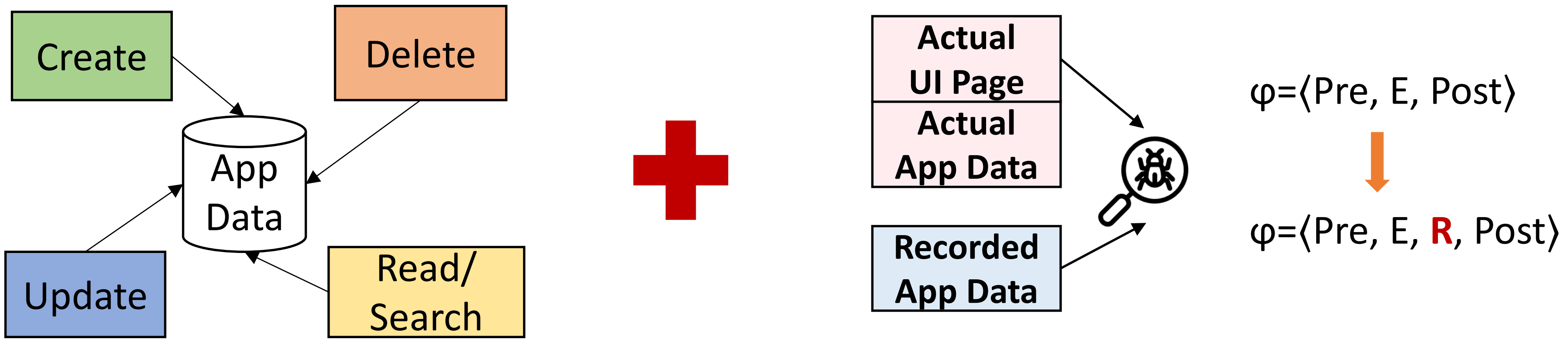
- Randomly ***interleave the relevant DMFs*** on the shared app data *(with other possible random events)* to exhibit *diverse* app states.



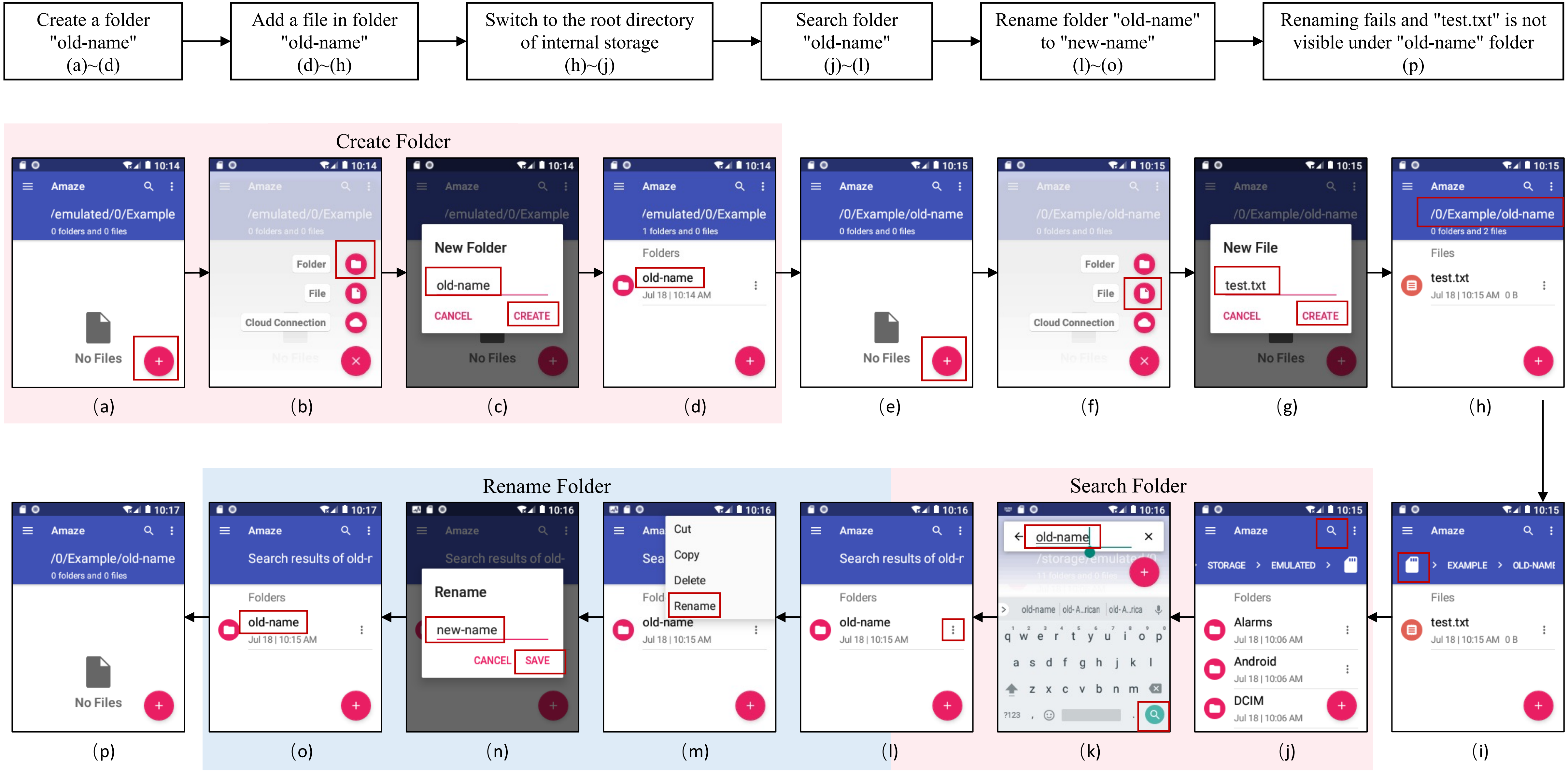
- Use ***an abstract data model*** to simulate the data update effect of E, thus facilitating property checking; and use the consistency between the app data and UI layouts to check the property.

How to Effectively *Check* the Properties of DMFs?

- Randomly ***interleave the relevant DMFs*** on the shared app data *(with other possible random events)* to exhibit *diverse* app states.



- Use ***an abstract data model*** to simulate the data update effect of E, thus facilitating property checking; and use the consistency between the app data and UI layouts to check the property.

</br>

Create a folder "old-name" (a)~(d)

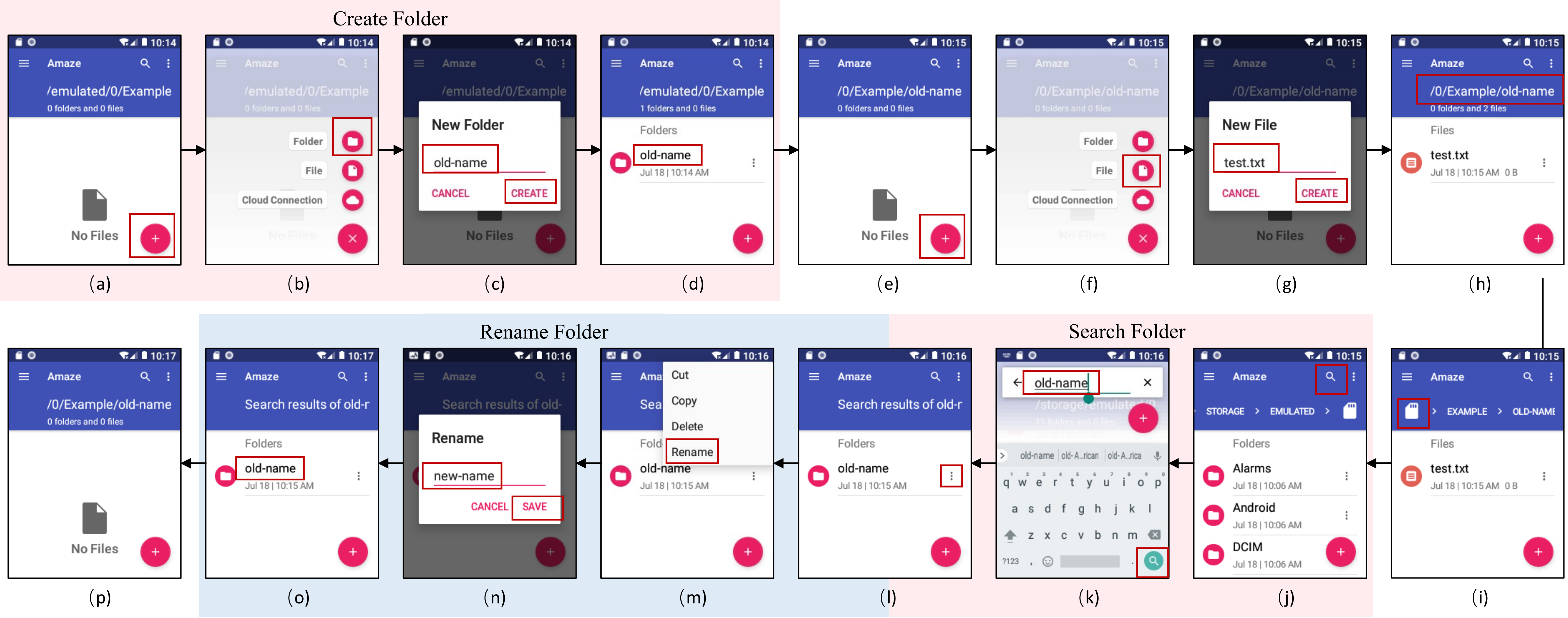
Add a file in folder "old-name" (d)~(h)

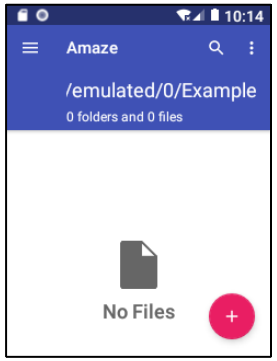
Switch to the root directory of internal storage (h)~(j)

Search folder "old-name" (j)~(l)

Rename folder "old-name" to "new-name" (l)~(o)

Renaming fails and "test.txt" is not visible under "old-name" folder (p)


Create Folder
Search Folder
Rename Folder
Amaze
/emulated/0/Example
0 folders and 0 files
No Files
Folder
File
Cloud Connection
New Folder
old-name
CANCEL
CREATE
1 folders and 0 files
Folders
Jul 18 | 10:14 AM
/0/Example/old-name
New File
test.txt
0 folders and 2 files
Files
Jul 18 | 10:15 AM 0 B
EXAMPLE
OLD-NAME
STORAGE
EMULATED
Alarms
Android
DCIM
Jul 18 | 10:06 AM
Search results of old-r
Jul 18 | 10:15 AM
Cut
Copy
Delete
Rename
new-name
SAVE
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)
(m)
(n)
(o)
(p)



10:14

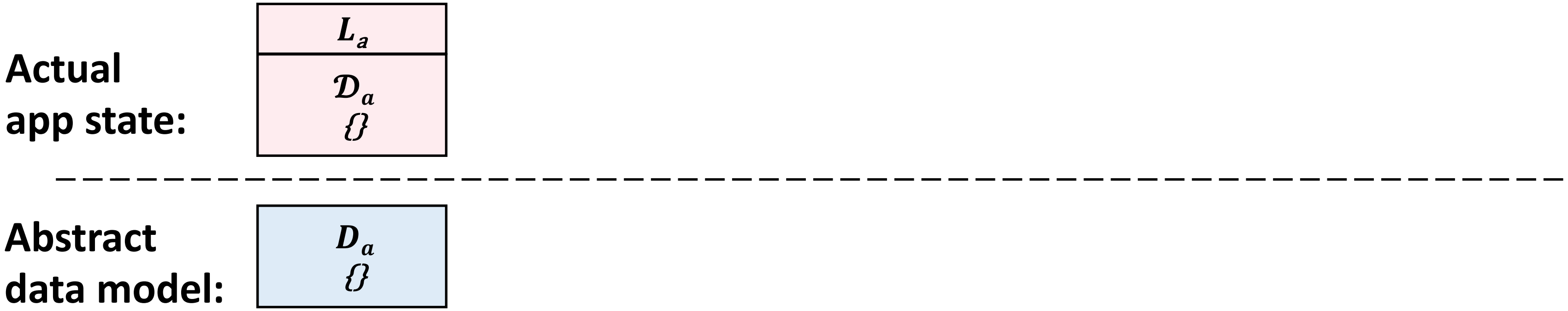
Amaze

/emulated/0/Example

0 folders and 0 files

No Files

(a)



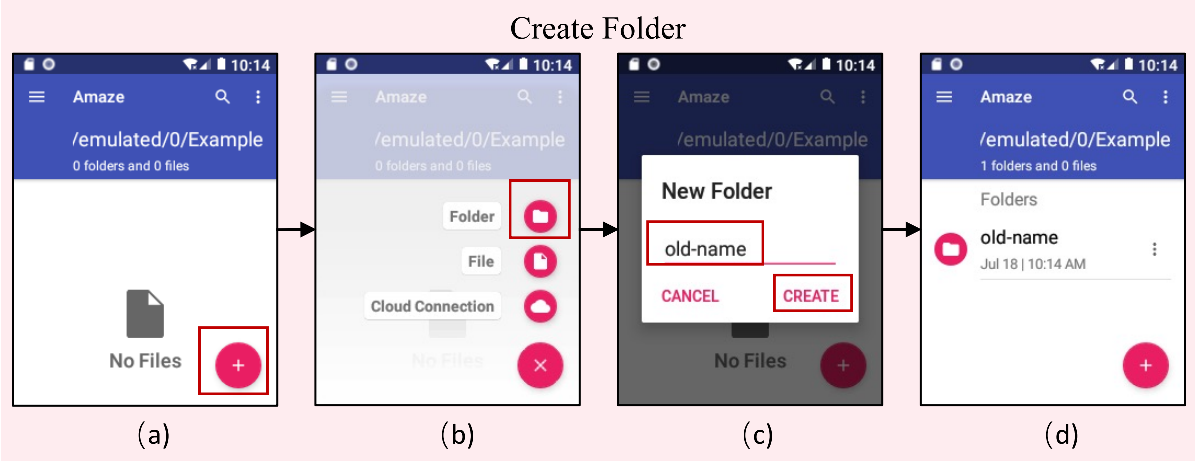
Actual app state:

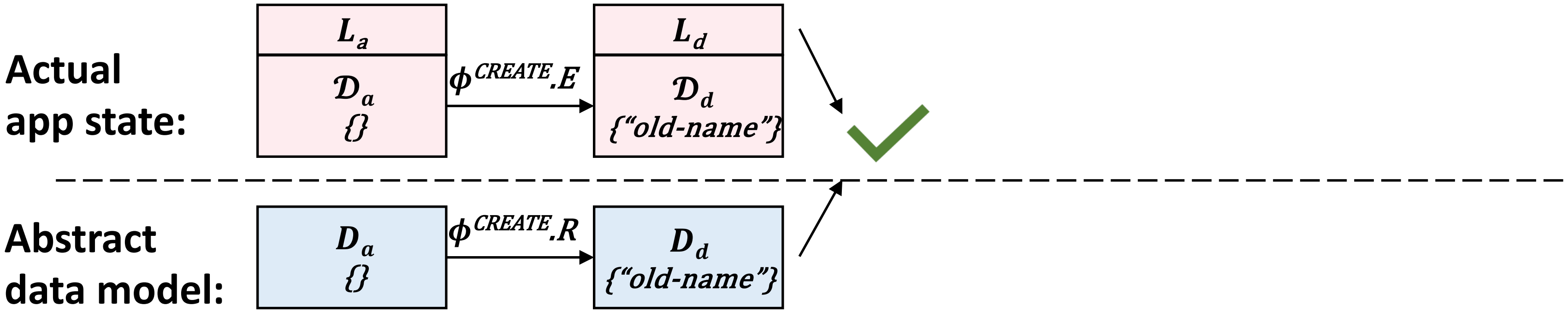
L_a

$\mathcal{D}_a$
$\{\}$

Abstract data model:

D_a
$\{\}$



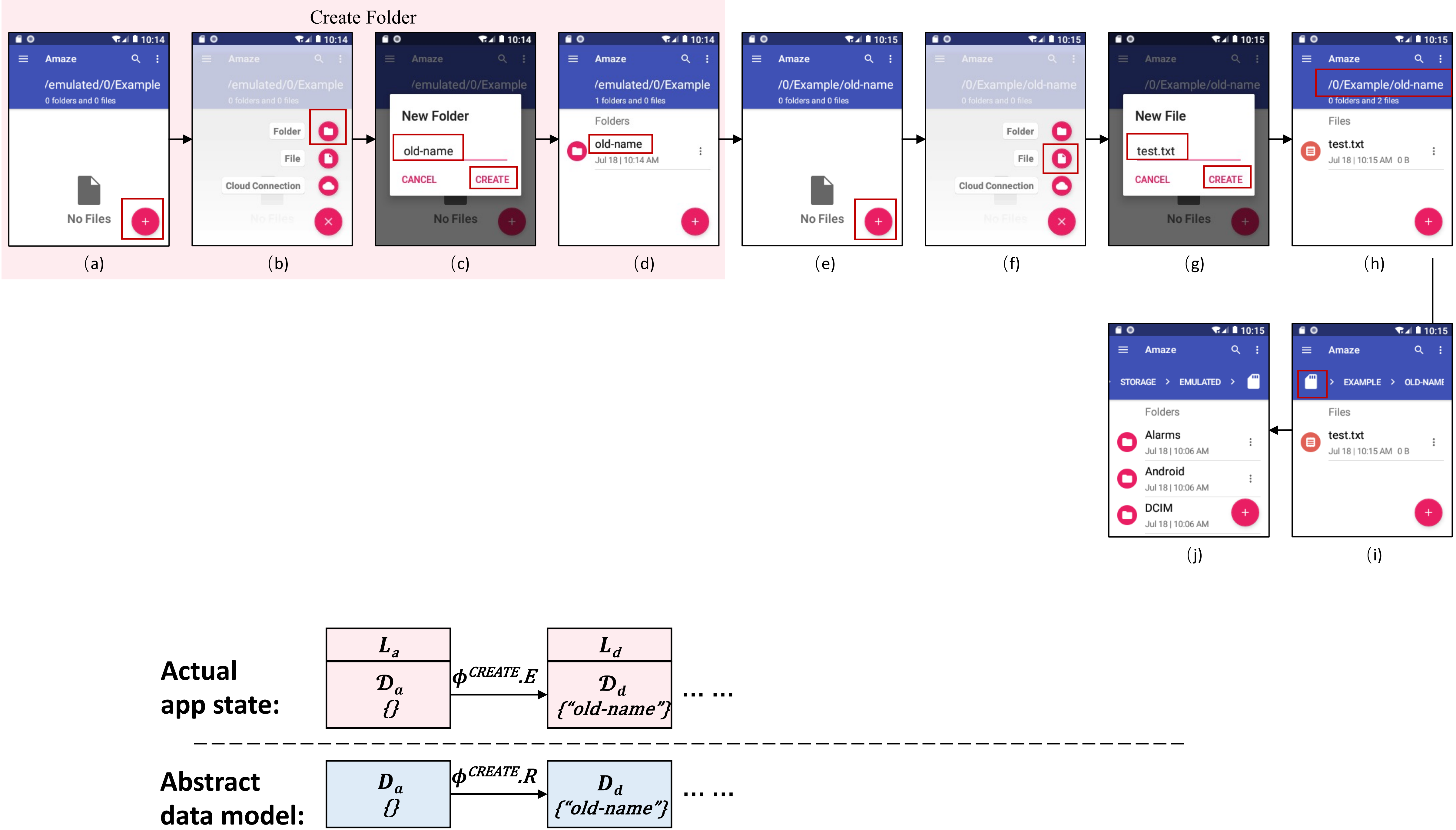


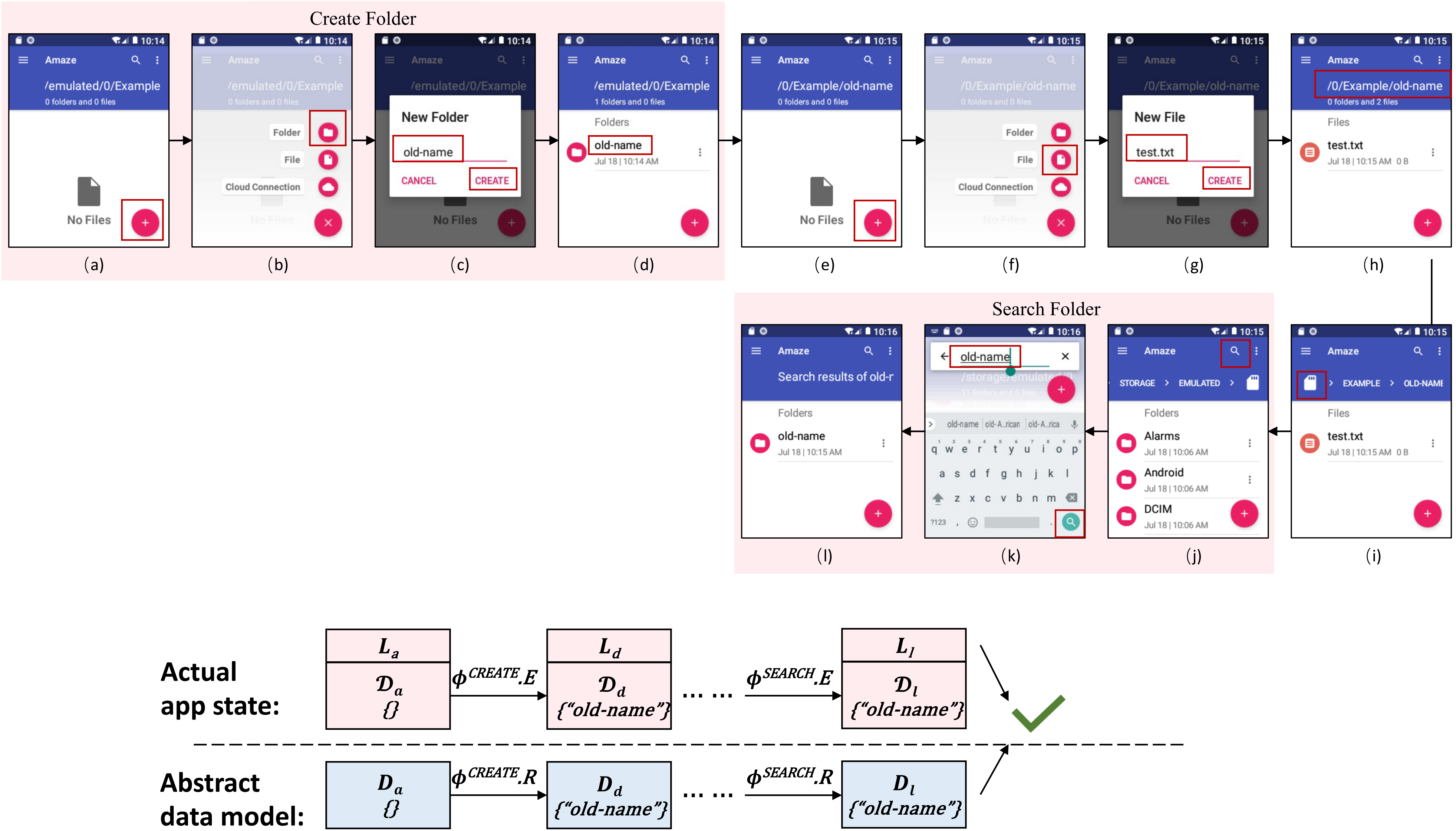
Actual app state:

L_a / $\mathcal{D}_a$ $\{\}$ $\xrightarrow{\phi^{CREATE}.E}$ L_d / $\mathcal{D}_d$ $\{\text{“old-name”}\}$ ✓

Abstract data model:

D_a $\{\}$ $\xrightarrow{\phi^{CREATE}.R}$ D_d $\{\text{“old-name”}\}$


Create Folder
Amaze
/emulated/0/Example
0 folders and 0 files
No Files
Folder
File
Cloud Connection
New Folder
old-name
CANCEL
CREATE
1 folders and 0 files
Folders
Jul 18 | 10:14 AM
/0/Example/old-name
New File
test.txt
0 folders and 2 files
Files
Jul 18 | 10:15 AM 0 B
STORAGE > EMULATED >
> EXAMPLE > OLD-NAME
Alarms
Android
DCIM
Jul 18 | 10:06 AM
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
Actual app state:
L_a
$\mathcal{D}_a$ {}
$\phi^{CREATE}.E$
L_d
$\mathcal{D}_d$ {“old-name”}
… …
Abstract data model:
D_a {}
$\phi^{CREATE}.R$
D_d {“old-name”}
… …

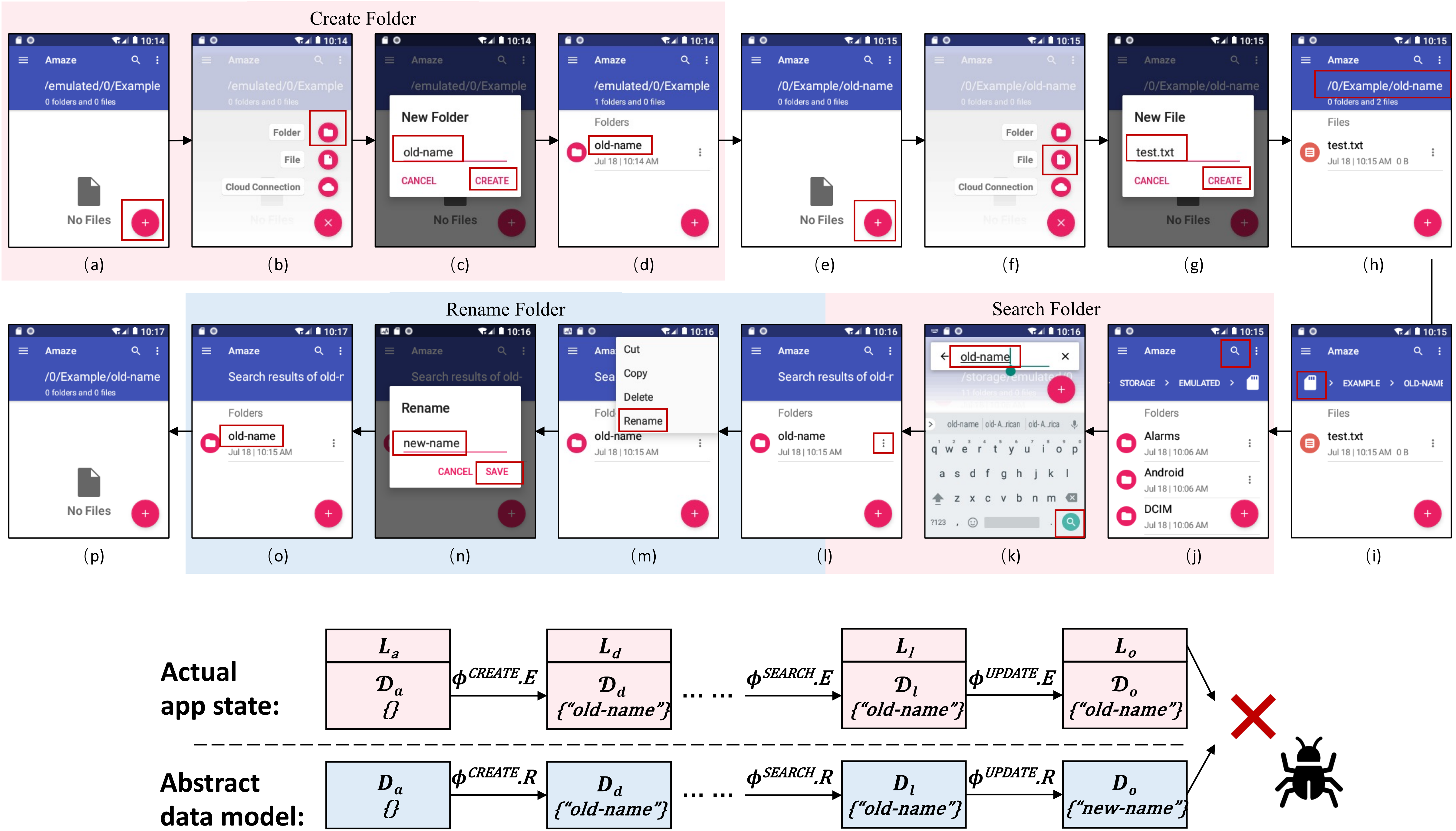


Actual app state:

L_a $\mathcal{D}_a$ $\{\}$ $\xrightarrow{\phi^{CREATE}.E}$ L_d $\mathcal{D}_d$ $\{\text{“old-name”}\}$ … … $\xrightarrow{\phi^{SEARCH}.E}$ L_l $\mathcal{D}_l$ $\{\text{“old-name”}\}$ ✓

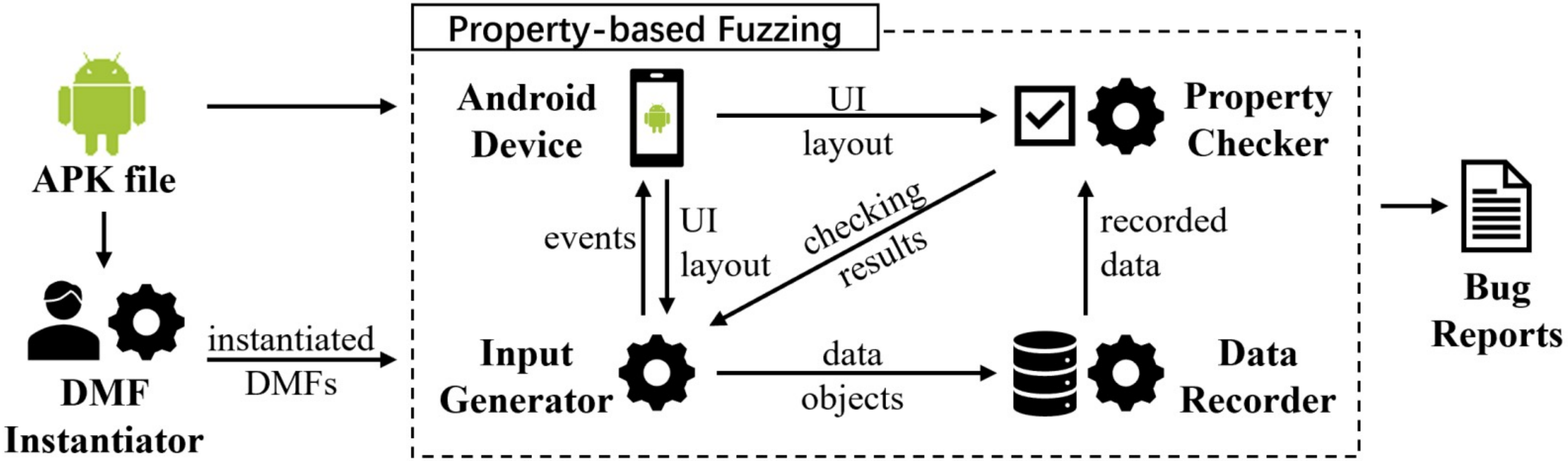
Abstract data model:

D_a $\{\}$ $\xrightarrow{\phi^{CREATE}.R}$ D_d $\{\text{“old-name”}\}$ … … $\xrightarrow{\phi^{SEARCH}.R}$ D_l $\{\text{“old-name”}\}$

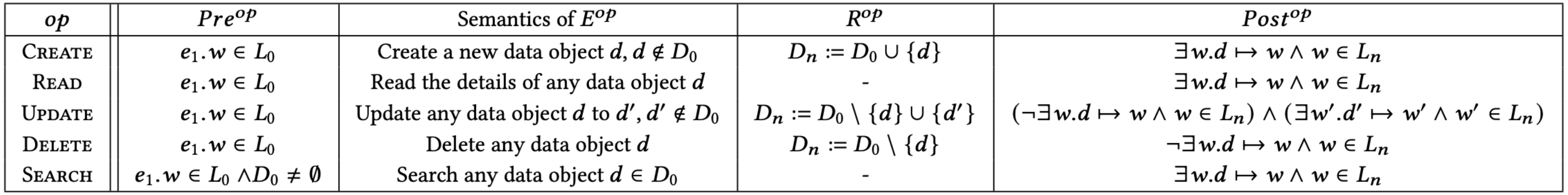


Create Folder
Search Folder
Rename Folder
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)
(m)
(n)
(o)
(p)
Amaze
/emulated/0/Example
0 folders and 0 files
No Files
Folder
File
Cloud Connection
New Folder
old-name
CANCEL
CREATE
1 folders and 0 files
Folders
Jul 18 | 10:14 AM
/0/Example/old-name
New File
test.txt
0 folders and 2 files
Files
Jul 18 | 10:15 AM 0 B
EXAMPLE
OLD-NAME
STORAGE
EMULATED
Alarms
Android
DCIM
Jul 18 | 10:06 AM
Search results of old-r
Cut
Copy
Delete
Rename
new-name
SAVE
Actual app state:
Abstract data model:
L_a
$\mathcal{D}_a$ {}
$\phi^{CREATE}.E$
L_d
$\mathcal{D}_d$ {"old-name"}
… …
$\phi^{SEARCH}.E$
L_l
$\mathcal{D}_l$ {"old-name"}
$\phi^{UPDATE}.E$
L_o
$\mathcal{D}_o$ {"old-name"}
D_a {}
$\phi^{CREATE}.R$
D_d {"old-name"}
$\phi^{SEARCH}.R$
D_l {"old-name"}
$\phi^{UPDATE}.R$
D_o {"new-name"}

Workflow of Our Approach



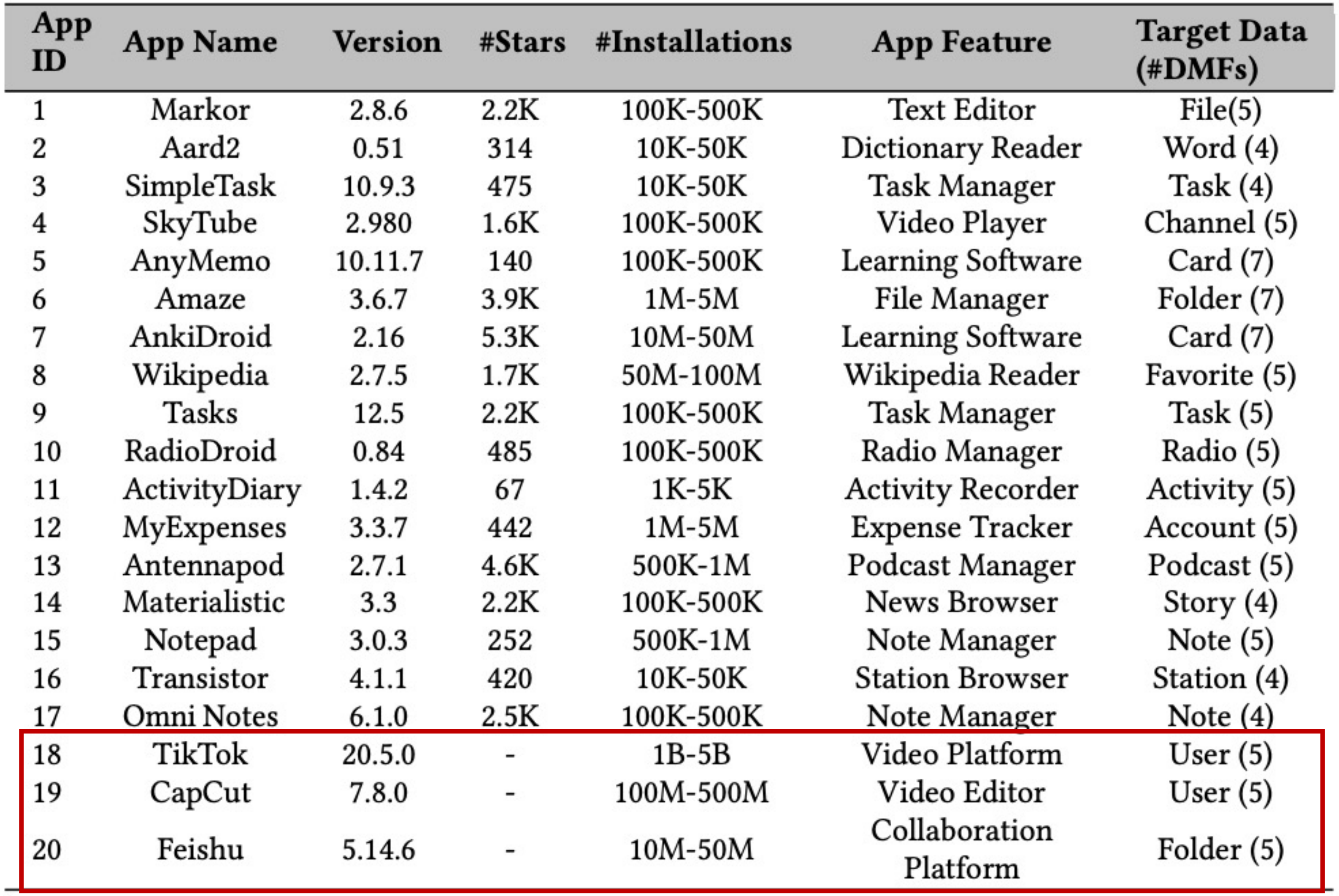
*abstract data model



op	Pre^{op}	Semantics of E^{op}	R^{op}	$Post^{op}$
Create	$e_1.w \in L_0$	Create a new data object $d, d \notin D_0$	$D_n := D_0 \cup \{d\}$	$\exists w.d \mapsto w \wedge w \in L_n$
Read	$e_1.w \in L_0$	Read the details of any data object d	-	$\exists w.d \mapsto w \wedge w \in L_n$
Update	$e_1.w \in L_0$	Update any data object d to $d', d' \notin D_0$	$D_n := D_0 \setminus \{d\} \cup \{d'\}$	$(\neg\exists w.d \mapsto w \wedge w \in L_n) \wedge (\exists w'.d' \mapsto w' \wedge w' \in L_n)$
Delete	$e_1.w \in L_0$	Delete any data object d	$D_n := D_0 \setminus \{d\}$	$\neg\exists w.d \mapsto w \wedge w \in L_n$
Search	$e_1.w \in L_0 \wedge D_0 \neq \emptyset$	Search any data object $d \in D_0$	-	$\exists w.d \mapsto w \wedge w \in L_n$

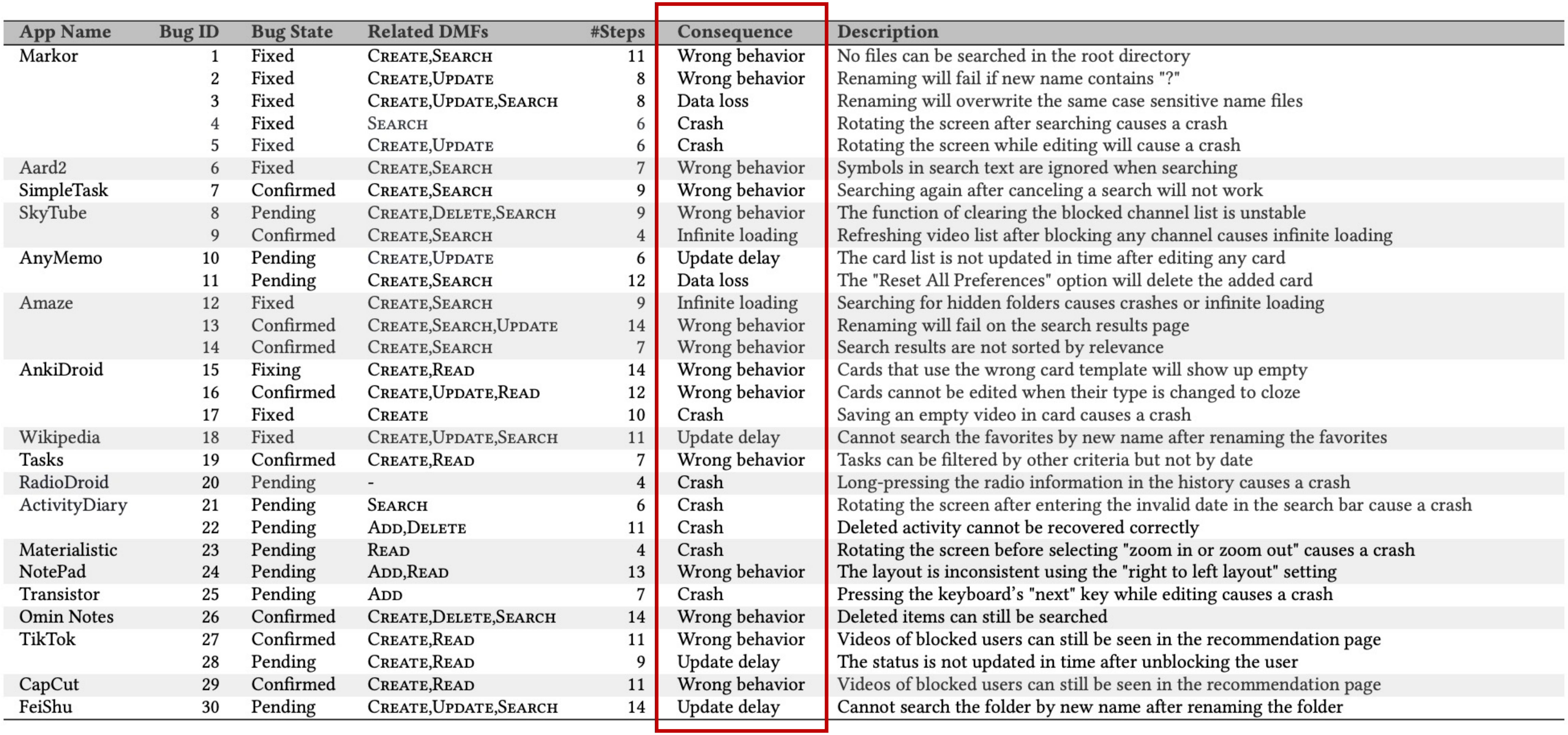
Evaluated Open-source and Industrial Apps

- ✓ **Selected 20 apps**
- ✓ **17 open-source apps**
- ✓ **3 industrial apps**



App ID	App Name	Version	#Stars	#Installations	App Feature	Target Data (#DMFs)
1	Markor	2.8.6	2.2K	100K-500K	Text Editor	File(5)
2	Aard2	0.51	314	10K-50K	Dictionary Reader	Word (4)
3	SimpleTask	10.9.3	475	10K-50K	Task Manager	Task (4)
4	SkyTube	2.980	1.6K	100K-500K	Video Player	Channel (5)
5	AnyMemo	10.11.7	140	100K-500K	Learning Software	Card (7)
6	Amaze	3.6.7	3.9K	1M-5M	File Manager	Folder (7)
7	AnkiDroid	2.16	5.3K	10M-50M	Learning Software	Card (7)
8	Wikipedia	2.7.5	1.7K	50M-100M	Wikipedia Reader	Favorite (5)
9	Tasks	12.5	2.2K	100K-500K	Task Manager	Task (5)
10	RadioDroid	0.84	485	100K-500K	Radio Manager	Radio (5)
11	ActivityDiary	1.4.2	67	1K-5K	Activity Recorder	Activity (5)
12	MyExpenses	3.3.7	442	1M-5M	Expense Tracker	Account (5)
13	Antennapod	2.7.1	4.6K	500K-1M	Podcast Manager	Podcast (5)
14	Materialistic	3.3	2.2K	100K-500K	News Browser	Story (4)
15	Notepad	3.0.3	252	500K-1M	Note Manager	Note (5)
16	Transistor	4.1.1	420	10K-50K	Station Browser	Station (4)
17	Omni Notes	6.1.0	2.5K	100K-500K	Note Manager	Note (4)
18	TikTok	20.5.0	-	1B-5B	Video Platform	User (5)
19	CapCut	7.8.0	-	100M-500M	Video Editor	User (5)
20	Feishu	5.14.6	-	10M-50M	Collaboration Platform	Folder (5)

Bug Finding Results on the Apps



App Name	Bug ID	Bug State	Related DMFs	#Steps	Consequence	Description
Markor	1	Fixed	Create,Search	11	Wrong behavior	No files can be searched in the root directory
	2	Fixed	Create,Update	8	Wrong behavior	Renaming will fail if new name contains "?"
	3	Fixed	Create,Update,Search	8	Data loss	Renaming will overwrite the same case sensitive name files
	4	Fixed	Search	6	Crash	Rotating the screen after searching causes a crash
	5	Fixed	Create,Update	6	Crash	Rotating the screen while editing will cause a crash
Aard2	6	Fixed	Create,Search	7	Wrong behavior	Symbols in search text are ignored when searching
SimpleTask	7	Confirmed	Create,Search	9	Wrong behavior	Searching again after canceling a search will not work
SkyTube	8	Pending	Create,Delete,Search	9	Wrong behavior	The function of clearing the blocked channel list is unstable
	9	Confirmed	Create,Search	4	Infinite loading	Refreshing video list after blocking any channel causes infinite loading
AnyMemo	10	Pending	Create,Update	6	Update delay	The card list is not updated in time after editing any card
	11	Pending	Create,Search	12	Data loss	The "Reset All Preferences" option will delete the added card
Amaze	12	Fixed	Create,Search	9	Infinite loading	Searching for hidden folders causes crashes or infinite loading
	13	Confirmed	Create,Search,Update	14	Wrong behavior	Renaming will fail on the search results page
	14	Confirmed	Create,Search	7	Wrong behavior	Search results are not sorted by relevance
AnkiDroid	15	Fixing	Create,Read	14	Wrong behavior	Cards that use the wrong card template will show up empty
	16	Confirmed	Create,Update,Read	12	Wrong behavior	Cards cannot be edited when their type is changed to cloze
	17	Fixed	Create	10	Crash	Saving an empty video in card causes a crash
Wikipedia	18	Fixed	Create,Update,Search	11	Update delay	Cannot search the favorites by new name after renaming the favorites
Tasks	19	Confirmed	Create,Read	7	Wrong behavior	Tasks can be filtered by other criteria but not by date
RadioDroid	20	Pending	-	4	Crash	Long-pressing the radio information in the history causes a crash
ActivityDiary	21	Pending	Search	6	Crash	Rotating the screen after entering the invalid date in the search bar cause a crash
	22	Pending	Add,Delete	11	Crash	Deleted activity cannot be recovered correctly
Materialistic	23	Pending	Read	4	Crash	Rotating the screen before selecting "zoom in or zoom out" causes a crash
NotePad	24	Pending	Add,Read	13	Wrong behavior	The layout is inconsistent using the "right to left layout" setting
Transistor	25	Pending	Add	7	Crash	Pressing the keyboard's "next" key while editing causes a crash
Omin Notes	26	Confirmed	Create,Delete,Search	14	Wrong behavior	Deleted items can still be searched
TikTok	27	Confirmed	Create,Read	11	Wrong behavior	Videos of blocked users can still be seen in the recommendation page
	28	Pending	Create,Read	9	Update delay	The status is not updated in time after unblocking the user
CapCut	29	Confirmed	Create,Read	11	Wrong behavior	Videos of blocked users can still be seen in the recommendation page
FeiShu	30	Pending	Create,Update,Search	14	Update delay	Cannot search the folder by new name after renaming the folder

* We found 30 bugs, 29 of which are DMEs (22 are non-crashing failures, and 7 are crashing ones).

Lessons Learned

Lessons Learned

Lesson learned

Property-based testing requires specifying the interested properties of the software under test for validation.

Strengths of PBT

- PBT can generate a large volume of tests with minimal human oversight and cost.
- PBT encourages us to make our assumptions explicit by defining properties.
- PBT can find things that a human tester does not think of.

Challenges of PBT

- PBT's input domain is not systematic, so it is unlikely to test edges cases like boundary values unless explicitly told to, and is unlikely to achieve good coverage of the software unless specifically guided.

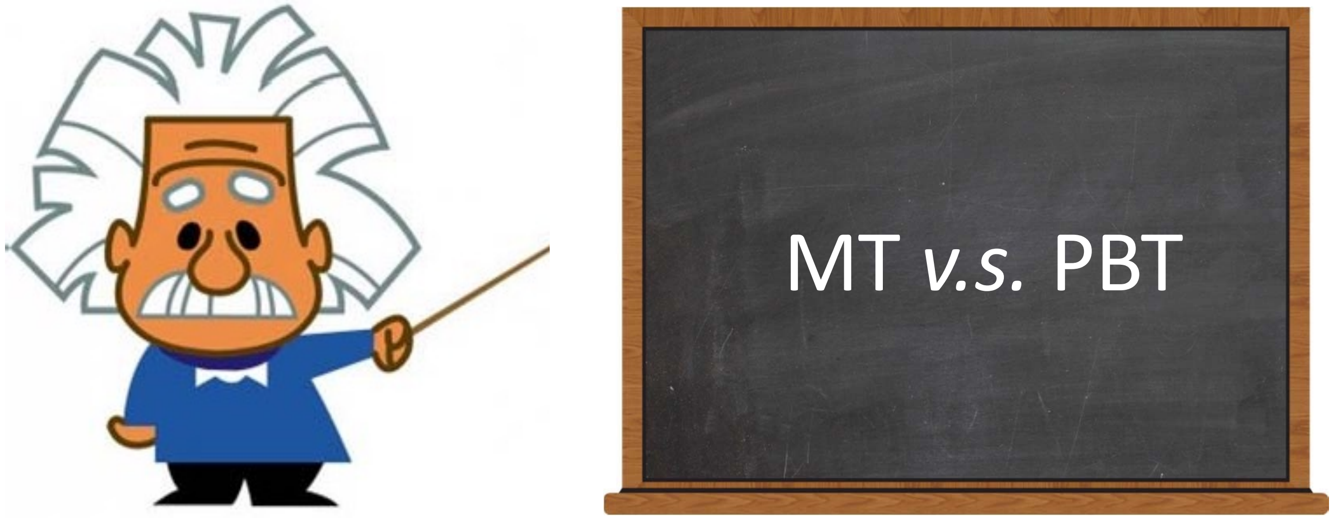
- PBT only tests the properties that it is given, which are (usually) partial properties. While specifying test oracles that are complete is possible, it is uncommon.

- Side-effects, dependencies, performance, …

Lessons Learned

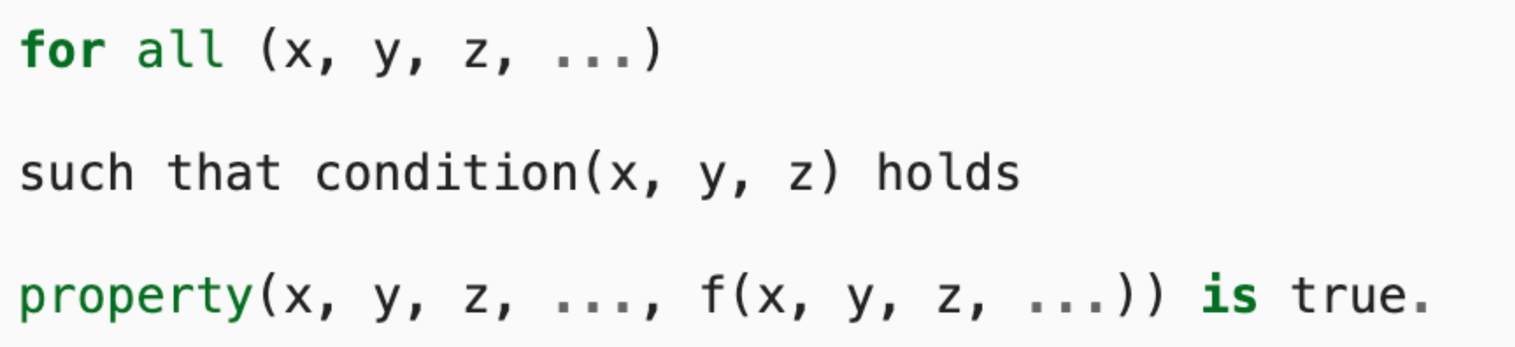
Lesson learned

Metamorphic testing (MT) is a special case of Property-based testing (PBT).



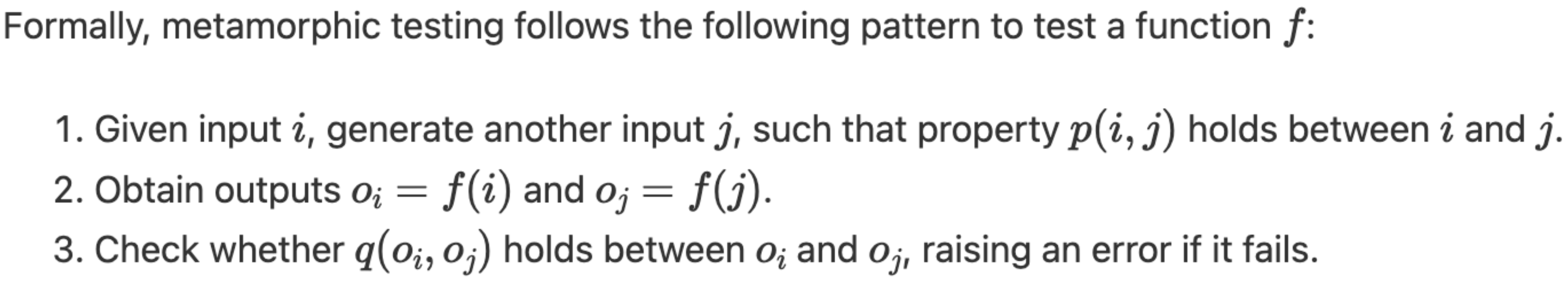
MT *v.s.* PBT

PBT



```
for all (x, y, z, ...)

such that condition(x, y, z) holds

property(x, y, z, ..., f(x, y, z, ...)) is true.
```

MT

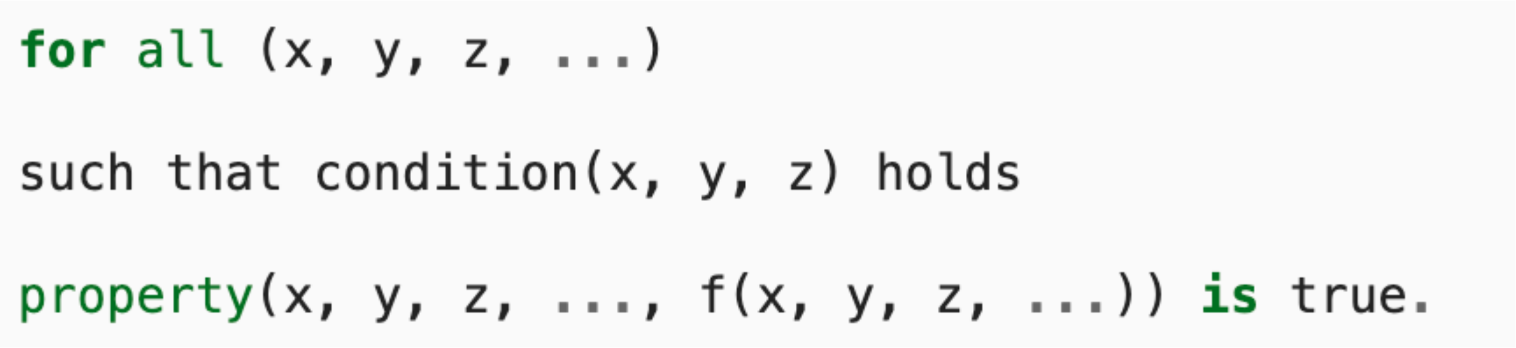


Formally, metamorphic testing follows the following pattern to test a function f:

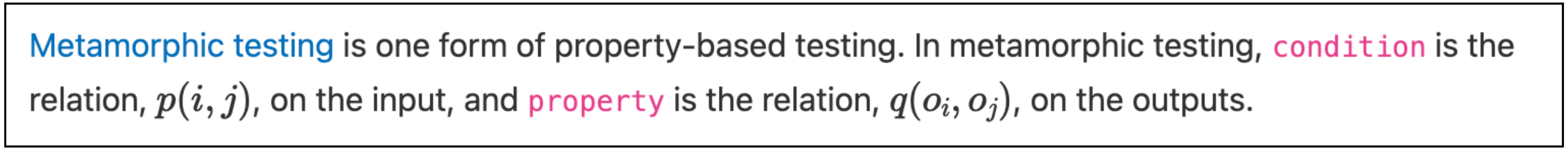
1. Given input i, generate another input j, such that property $p(i, j)$ holds between i and j.
2. Obtain outputs $o_i = f(i)$ and $o_j = f(j)$.
3. Check whether $q(o_i, o_j)$ holds between o_i and o_j, raising an error if it fails.

MT is one special form of PBT

PBT

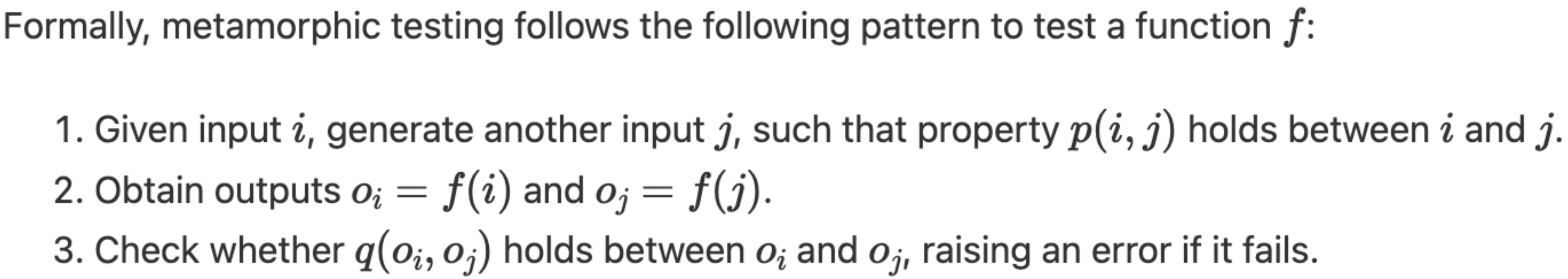


```
for all (x, y, z, ...)

such that condition(x, y, z) holds

property(x, y, z, ..., f(x, y, z, ...)) is true.
```



Metamorphic testing is one form of property-based testing. In metamorphic testing, `condition` is the relation, $p(i, j)$, on the input, and `property` is the relation, $q(o_i, o_j)$, on the outputs.

MT



Formally, metamorphic testing follows the following pattern to test a function f:

1. Given input i, generate another input j, such that property $p(i, j)$ holds between i and j.
2. Obtain outputs $o_i = f(i)$ and $o_j = f(j)$.
3. Check whether $q(o_i, o_j)$ holds between o_i and o_j, raising an error if it fails.

Lessons Learned

Lesson learned

There are five approaches to specifying the properties for PBT, and different properties have different fault revealing abilities.

How To Specify It (Property)!

❑Validity Testing

“Every operation should return valid results.”

❑Postconditions

“Postconditions relate return values to arguments of a single call.”

❑Metamorphic Properties

“Related calls return related results.”

❑Inductive Testing

“Inductive proofs inspire inductive tests.”

❑Model-based Properties

“Abstract away from details to simplify properties.”

*https://johanneslink.net/how-to-specify-it

How To Specify It (Property)!

❑Validity Testing

“Every operation should return valid results.”

❑Postconditions

“Postconditions relate return values to arguments of a single call.”

❑Metamorphic Properties

“Related calls return related results.”

❑Inductive Testing

“Inductive proofs inspire inductive tests.”

❑Model-based Properties

“Abstract away from details to simplify properties.”

*https://johanneslink.net/how-to-specify-it

总结和思考

- 如何利用领域知识解决Oracle问题？

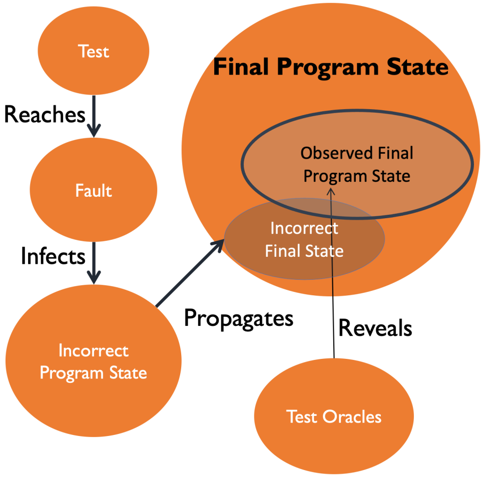
- 如何针对复杂软件和系统构造蜕变关系？如何针对复杂软件和系统构建基于属性的测试框架？

- 如何将有效的测试用例设计技术与上述Oracle技术结合？

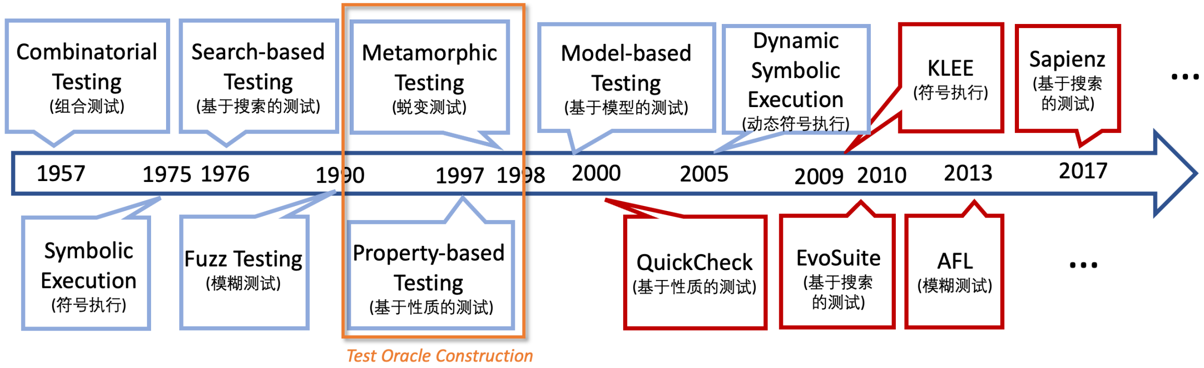
- 如何以更好的人机交互方式解决Oracle问题？

Software Fault and Failure Model (软件错误模型)

- Reachability
- Infection
- Propagation
- Revealability



Techniques for *Oracle Construction*



Combinatorial Testing (组合测试)
Search-based Testing (基于搜索的测试)
Metamorphic Testing (蜕变测试)
Model-based Testing (基于模型的测试)
Dynamic Symbolic Execution (动态符号执行)
KLEE (符号执行)
Sapienz (基于搜索的测试)
…

1957 1975 1976 1990 1997 1998 2000 2005 2009 2010 2013 2017

Symbolic Execution (符号执行)
Fuzz Testing (模糊测试)
Property-based Testing (基于性质的测试)
QuickCheck (基于性质的测试)
EvoSuite (基于搜索的测试)
AFL (模糊测试)
…

Test Oracle Construction

* First proposed, **Became popular.**

Metamorphic Testing Process

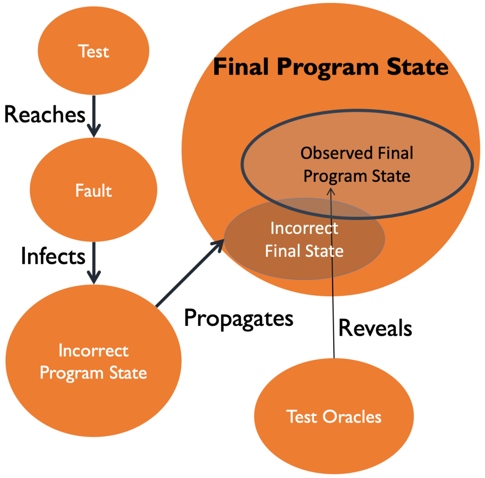
1. Identification of metamorphic relations. 根据领域知识**手工**分析识别
2. Generation/Selection of source test cases. **自动**随机生成/**复用**现有tests
3. Generation of follow-up test cases. **自动**生成
4. Checking of metamorphic relations. **自动**验证

Property-based Testing Pattern

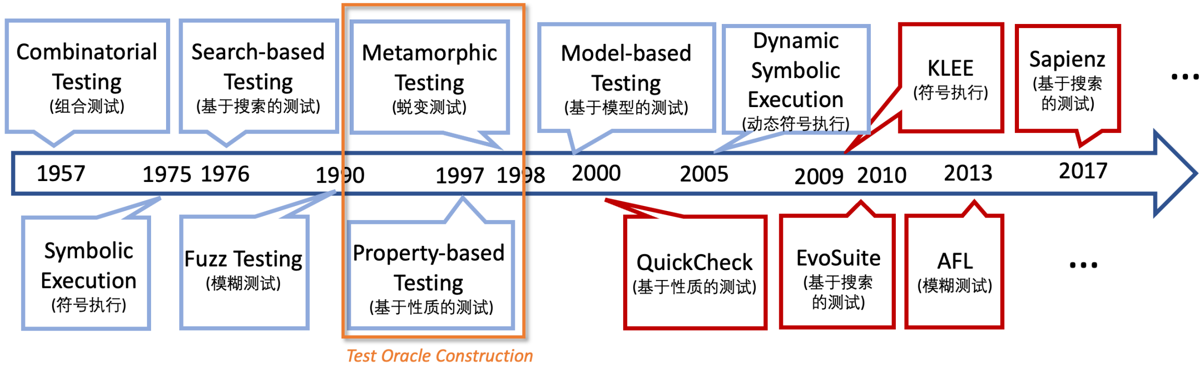
1. Generating random inputs **自动**生成随机输入
2. Ensuring the condition holds **自动**检查前置条件是否满足
3. Checking whether a property holds on those inputs **自动**检查性质是否成立
4. Shrinking the counter-example input to obtain the minimal one **自动**约减反例

Software Fault and Failure Model (软件错误模型)

- Reachability
- Infection
- Propagation
- Revealability



Techniques for *Oracle Construction*

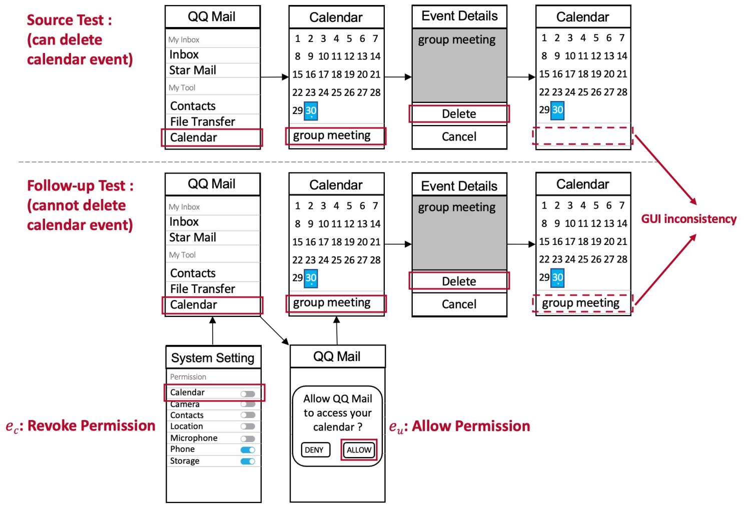

Combinatorial Testing (组合测试)
Search-based Testing (基于搜索的测试)
Metamorphic Testing (蜕变测试)
Model-based Testing (基于模型的测试)
Dynamic Symbolic Execution (动态符号执行)
KLEE (符号执行)
Sapienz (基于搜索的测试)
...
1957
1975
1976
1990
1997
1998
2000
2005
2009
2010
2013
2017
Symbolic Execution (符号执行)
Fuzz Testing (模糊测试)
Property-based Testing (基于性质的测试)
QuickCheck (基于性质的测试)
EvoSuite (基于搜索的测试)
AFL (模糊测试)
...
Test Oracle Construction

* First proposed, Became popular.

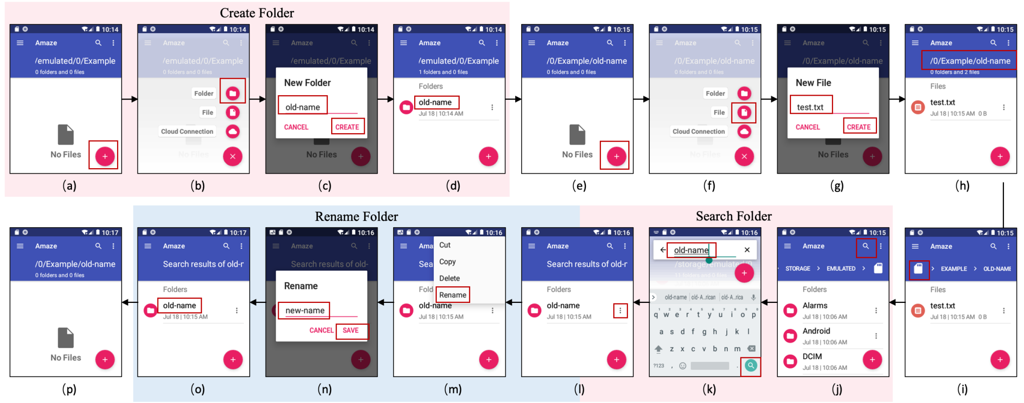
System Setting-related Defects

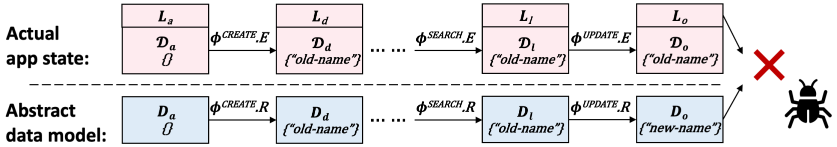
QQ Mail

4,500,000,000+
installations in major
app markets



Data Manipulation (CRUD) Errors


Create Folder
Rename Folder
Search Folder
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)
(m)
(n)
(o)
(p)
Amaze
/emulated/0/Example
New Folder
old-name
CANCEL
CREATE
/0/Example/old-name
New File
test.txt
No Files
Rename
new-name
SAVE


Actual app state:
L_a
$\mathcal{D}_a$ {}
$\phi^{CREATE}.E$
L_d
$\mathcal{D}_d$ {"old-name"}
... ...
$\phi^{SEARCH}.E$
L_l
$\mathcal{D}_l$ {"old-name"}
$\phi^{UPDATE}.E$
L_o
$\mathcal{D}_o$ {"old-name"}
Abstract data model:
D_a {}
$\phi^{CREATE}.R$
D_d {"old-name"}
... ...
$\phi^{SEARCH}.R$
D_l {"old-name"}
$\phi^{UPDATE}.R$
D_o {"new-name"}