

Towards Trustworthy Program Repair

Yingfei Xiong, Peking University,
Dagstuhl Seminar, 2017

State of the Art

- Precision = $\frac{\#Correctly\ Repaired\ Defects}{\#All\ Defects\ with\ Patches}$
- Recall = $\frac{\#Correctly\ Repaired\ Defects}{\#All\ Defects}$

Name	Precision	Recall
Prophet	38.5%*	14.3%
Angelix	35.7%	12.2%
HistoricalFix	--	6.4%

* Only the first patch is considered if multiple are generated

State of the Art

From the review of a top SE conference

The presented approach achieves a precision of over 70% which is significantly higher than precisions of previous approaches. The recall figures, however, are low as for all the other approaches... This shows that the research on automatic program repair **is still in its infancy**. The presented approach **does not change this situation** significantly.

Decision: Rejection

Crisis of Program Repair

- The precision and recall are too low to promise a bright future of program repair
- If we cannot change the situation significantly, we shall be neglected by the academia and industry

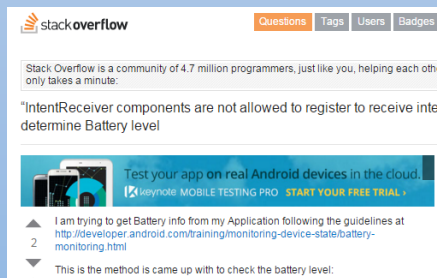
Overview

Is it possible to improve precision and recall?

A Study of manual repair [Submitted]

How to improve precision?

Learn from QA sites
[ASE15]



Precise Condition Synthesis
[ICSE17]



[Submitted] Jiajun Jiang, Yingfei Xiong. A Case Study on Manual Repair of Program Defects. Submitted.

[ASE15] Qing Gao, Hansheng Zhang, Jie Wang, Yingfei Xiong, Lu Zhang, Hong Mei. Fixing Recurring Crash Bugs via Analyzing Q&A Sites. ASE'15

[ICSE17] Yingfei Xiong, Jie Wang, Runfa Yan, Jiachen Zhang, Shi Han, Gang Huang, Lu Zhang. Precise Condition Synthesis for Program Repair. ICSE'17

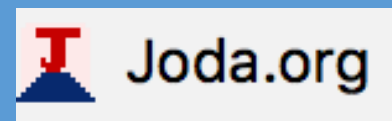
RQ1: Is it possible to improve precision and recall?

- A developer manually repaired 50 defects from Defects4J
- The settings are the same as repair tools
- If high precision and recall is achieved
 - There is hope
 - There is a way

Settings

- The developer has access to the source code and the tests
- The developer is not familiar with the projects
- The developer cannot read the documents
- The developer cannot search project-specific information on the Internet

Defects4J



Findings and Implications

Precision: 89.4% Recall: 84.0%

- A lot of rooms to improve under current settings

Reason of Imperfection: the lack of domain knowledge

- We need to break the settings at some point

Average fixing time: 47 minutes

- Automatic program repair is valuable

Manual repair can be summarized into strategies

- Possibility for improving automatic techniques

No single strategy dominates the defects

- We need combine different strategies

RQ2: Can we improve precision?

- Why Precision?
 - Precision decides the relative cost of using the technique
 - Conjecture: Developers adopt a technique only when the relative cost is small, i.e., $\text{cost} \ll \text{benefit}$
- How to improve?
 - Learn the experience of the developers

Fixing from QA Sites

- How do developers get their experience?

```
29     public void onReceive (final Context context, final Intent intent) {
30         final int action = intent.getExtras().getInt(KEY_ACTION, -1);
31         final float bl = BatteryHelper.level(context);
32         LOG.i("AlarmReceiver invoked: action=%s bl=%s.", action, bl);
33         switch (action) {
34             ...
35             ...
36             ...
37             ...
38             ...
39             ...
40             ...
41             ...
42             ...
43             ...
44             ...
45             ...
46             ...
47             ...
48             ...
49             ...
50             ...
51         }
52     }
```

java.lang.RuntimeException: Unable to start receiver
com.vaguehope.onosendai.update.AlarmReceiver:

Fixing from QA Sites

java.lang.RuntimeException: Unable to start receiver : android.conter

Web Videos News Images More Search tools

8 results (0.52 seconds)

android - "IntentReceiver components are not allowed to ...
stackoverflow.com/.../intentreceiver-components-are-not-allowed-to-regi...
Jul 24, 2014 - "IntentReceiver components are not allowed to register to receive ...
ACTION_BATTERY_CHANGED); Intent batteryStatus = c. ... RuntimeException:
Unable to start receiver ... ActivityThread.main(ActivityThread.java:4627) at java.
lang.reflect. ... NativeStart.main(Native Method) Caused by: android.content

android - Battery changed broadcast receiver crashing app ...
stackoverflow.com/.../battery-changed-broadcast-receiver-crashing-app-...
Feb 27, 2013 - Battery changed broadcast receiver crashing app on some phones. No
... PowerConnectionReceiver"> <intent-filter> <action android:name="android.intent
.action. ... RuntimeException: Unable to start receiver com.doublep.wakey.
ReceiverCallNotAllowedException: IntentReceiver components are not ...

android - Want app to execute some code when phone is ...
stackoverflow.com/.../want-app-to-execute-some-code-when-phone-is-pl...
Jun 29, 2012 - ACTION_BATTERY_CHANGED)); int plugged = intent. ... The code
errors out with: *FATAL EXCEPTION: main: java.lang.RuntimeException: Unable to
start receiver com.example.CharainaOnReceiver: android.content. ... IntentReceiver



Questions Tags Users Badges

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other. It only takes a minute:

"IntentReceiver components are not allowed to register to receive inter
determine Battery level

Test your app on real Android devices in the cloud.
keynote MOBILE TESTING PRO START YOUR FREE TRIAL

I am trying to get Battery info from my Application following the guidelines at
<http://developer.android.com/training/monitoring-device-state/battery-monitoring.html>

This is the method is came up with to check the battery level:

```
public void sendBatteryInfoMessage(){  
  
    IntentFilter iFilter = new IntentFilter(Intent.ACTION_BATTERY_  
    Intent batteryStatus = c.registerReceiver(null, iFilter);
```

Challenge of Analyzing QA Sites

- It is hard to understand natural languages

▲ Instead of:

4

```
context.registerReceiver(null, new IntentFilter(Intent.ACTION_BATTERY_CHANGED));
```

▼ use:

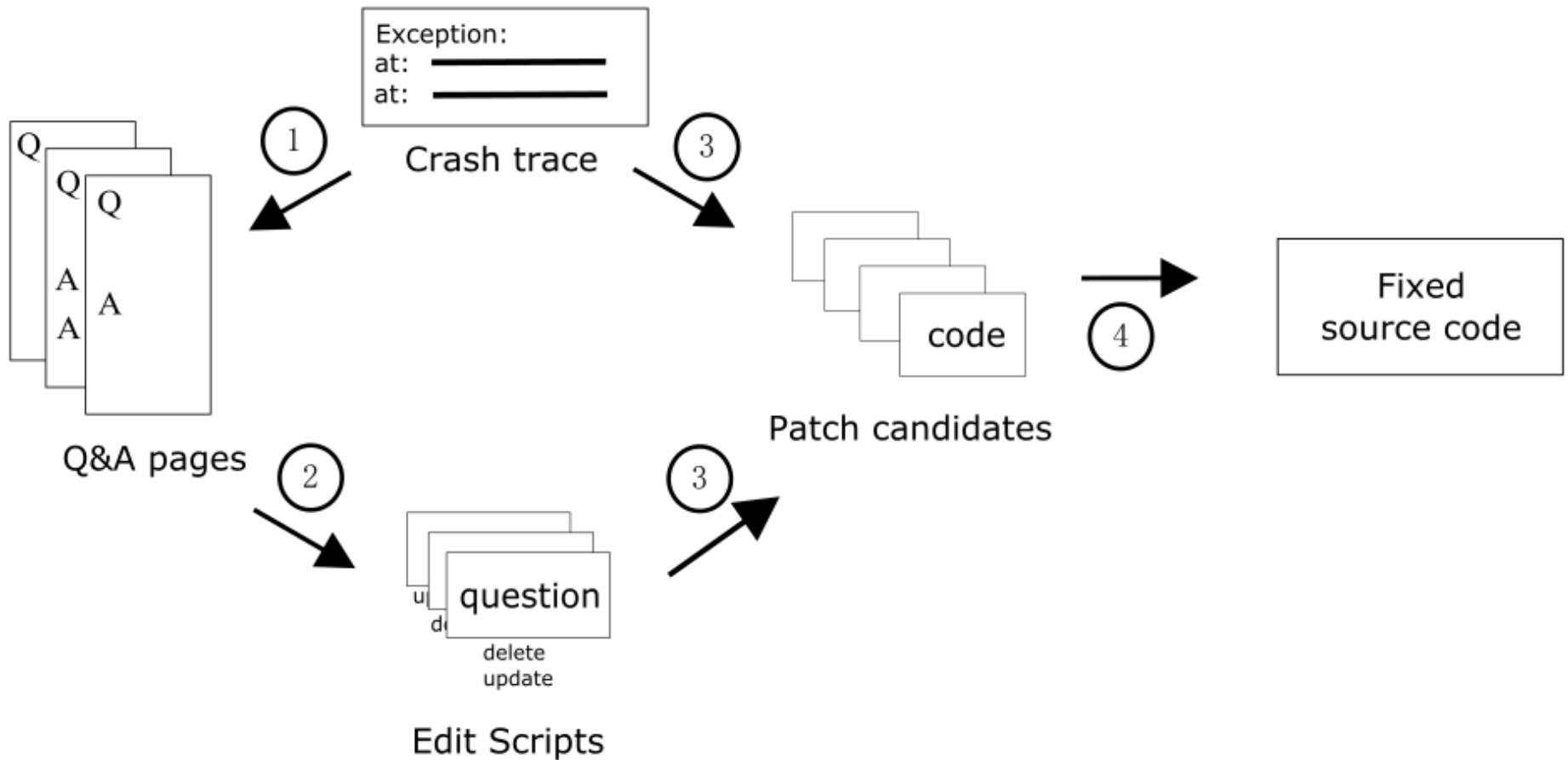
✓

```
context.getApplicationContext().registerReceiver(null, new IntentFilter(Intent.ACTION_BATTER
```

This is annoying -- `registerReceiver()` should be smarter than this -- but it's the workaround for this particular case.

- Observation: programmers communicates in programming languages
- Solution: Directly compare the code pieces

Approach Overview



Experiments

- 24 Android crash bugs that have answers on StackOverflow
 - Selected out of 161 Android crash bugs
- Correctly Fixed : 8
- Wrongly Fixed : 2
- Precision : 80%
- Recall : 33% (5% among Android crash bugs)

Precise Condition Synthesis

Condition bugs are very common

```
lcm = Math.abs(a+b);  
+ if (lcm == Integer.MIN_Value)  
+   throw new ArithmeticException();
```

Missing boundary checks

```
- if (hours <= 24)  
+ if (hours < 24)  
    withinOneDay=true;
```

Conditions too weak

```
- if (a > 0)  
+ if (a >= 0)  
    nat++;
```

Conditions too strong

Challenge – Incorrect Plausible Patches are Many

```
int lcm=Math.abs(  
    mulAndCheck(a/gdc(a,b),b));  
+if (lcm == Integer.MIN_VALUE) {  
+    throw new ArithmeticException();  
+}  
return lcm;
```

Test 1:

Input: a = 1, b = 50

Oracle: lcm = 50

Test 2:

Input: a = Integer.MIN_VALUE, b = 1

Oracle: Expected(ArithmeticException)

Exception-throwing Template:
If (\$C) throw \$E;

\$C: a synthesized condition

\$E: an expected exception

Correct condition:

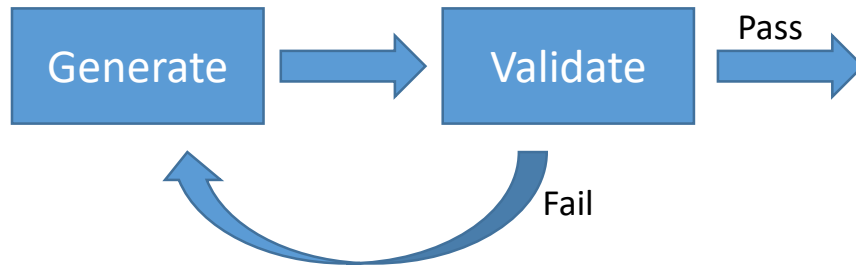
`lcm == Integer.MIN_VALUE`

Plausible conditions:

- `a > 1`
- `b == 1`
- `lcm != 50`
- ...

How to improve precision?

- The generate-validate framework



- Rank potential patches by their probabilities of being correct
- Existing ranking approaches are too coarse-grained for conditions
 - See paper for a discussion

Ranking Method 1: Rank Variables by Data-Dependency

- Locality of variable uses: recently assigned variables are more likely to be used
- Rank variable by data-dependency
 - $lcm = \text{Math.abs}(\text{mulAndCheck}(a/\text{gdc}(a, b), b))$
 - $lcm > a, lcm > b$

Ranking Method 2: Filter Variables by JavaDoc

```
/** ...  
 * @throws IllegalArgumentException if initial is not between  
 * min and max (even if it is a root)  
 **/
```

Only variable “initial” is considered when
throwing IllegalArgumentException

Ranking Method 3: Rank Predicates by Context

- The predicate tested on the variables are related to its context

Variable Type

```
Vector v = ...;  
if (v == null) return 0;
```

Variable Name

```
int hours = ...;  
if (hours < 24)  
    withinOneDay=true;
```

Method Name

```
int factorial() {  
    ...  
    if (n < 21) {  
        ...  
    }  
}
```

- Calculate the conditional probabilities from existing code base

ACS System

- ACS = Accurate Condition Synthesis
- Two sets of templates for repair

Oracle Returning

- Inserting one of the following statement before the statement causing failure
 - if (\$C) throw \$E;
 - if (\$C) return \$O;

Condition modifying

- Changing the condition located by predicate switching
 - if (\$D) => if (\$D || \$C)
 - if (\$D) => if (\$D && \$C)

Evaluation on Defects4J

Approach	Correct	Incorrect	Precision	Recall
ACS	18	5	78.3%	8.0%
jGenProg	5	22	18.5%	2.2%
Nopol	5	30	14.3%	2.2%
xPAR	3	- ⁴	- ⁴	1.3% ²
HistoricalFix ¹	10(16) ³	- ⁴	- ⁴	4.5%(7.1%) ^{2,3}

Conclusion

- Shall we change the current setting?
 - We still have a lot of rooms for improvement
- Can we improve precision?
 - Yes, at least for condition repair
- How can we improve precision?
 - By learning from existing resources