

進捗報告

仮テーマ「静的解析ツールで検出不可能なバグを含むクローンの特徴分析」

発表者：野口耕二郎

目次

- 背景, 目的
- アプローチ
- 進捗状況

背景，目的

背景

- バグがクローンに含まれている場合、クローンセット全体のクローン片を修正する必要がある
- クローンに含まれるバグを検出することができればバグを探す手間を省いたり、バグが拡散することを防いだりすることができる
- 静的解析ツールでクローンに含まれるバグを検出したいが、クローンに含まれるバグをメインに検出するバグカテゴリーは用意されていない

目的

- 現時点で静的解析ツールはクローンに含まれるバグをどれだけ検出することができるかを調べる
- 静的解析ツールによって検出できなかったバグにはどのような特徴があるかを分析する
- 得られた特徴からどのようにすれば静的解析ツールでクローンに含まれるバグを検出できるかを考える

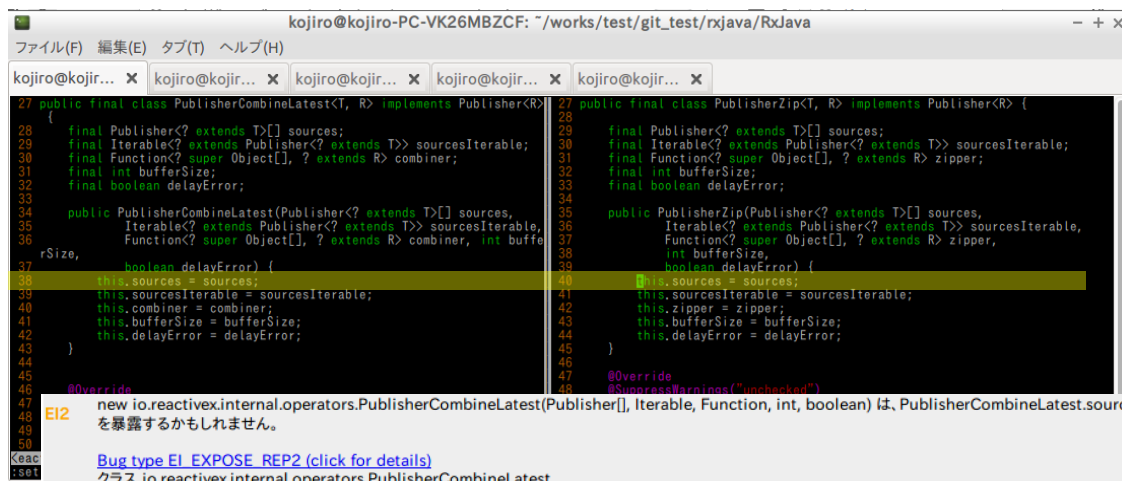
アプローチ

アプローチ

- FindBugsのバグを含むクローンを抽出する
- 実際のバグを含むクローンを抽出する
- FindBugsのバグと実際のバグの両方を含むクローンがどれだけ存在するかを調べる
- FindBugsでは検出できず実際のバグを含むクローンの特徴を分析する

FindBugsのバグを含むクローンの抽出

- クローンの範囲内にFindBugsが検出したバグの行が入っていればクローンの中にFindBugsが検出したバグが含まれているとする



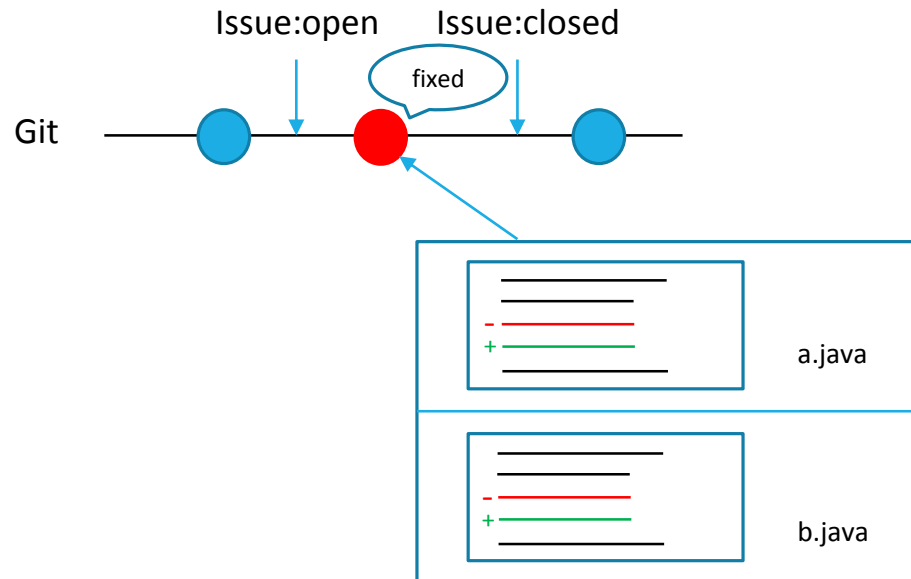
```
27 public final class PublisherCombineLatest<T, R> implements Publisher<R> {
28     {
29         final Publisher<? extends T>[] sources;
30         final Iterable<? extends Publisher<? extends T>> sourcesIterable;
31         final Function<? super Object[], ? extends R> combiner;
32         final int bufferSize;
33         final boolean delayError;
34     }
35     public PublisherCombineLatest(Publisher<? extends T>[] sources,
36         Iterable<? extends Publisher<? extends T>> sourcesIterable,
37         Function<? super Object[], ? extends R> combiner, int buffer
38         rSize,
39         boolean delayError) {
40         this.sources = sources;
41         this.sourcesIterable = sourcesIterable;
42         this.combiner = combiner;
43         this.bufferSize = bufferSize;
44         this.delayError = delayError;
45     }
46     @Override
47     new io.reactivex.internal.operators.PublisherCombineLatest(Publisher[], Iterable, Function, int, boolean)
48     を暴露するかもしれません。
49
50 Bug type EI_EXPOSE_REP2 (click for details)
51 クラス io.reactivex.internal.operators.PublisherCombineLatest
52 メソッド new io.reactivex.internal.operators.PublisherCombineLatest(Publisher[], Iterable, Function, int, boolean)
53 フィールド io.reactivex.internal.operators.PublisherCombineLatest.sources
54 ローカル変数名 sources
55 該当箇所 PublisherCombineLatest.java (line 38)
```

EI2 new io.reactivex.internal.operators.PublisherZip(Publisher[], Iterable, Function, int, boolean) は、PublisherZip.sources の中に外部の可変オブジェクトを格納することによって内部表現を暴露するかもしれません。

Bug type EI_EXPOSE_REP2 (click for details)
クラス io.reactivex.internal.operators.PublisherZip
メソッド new io.reactivex.internal.operators.PublisherZip(Publisher[], Iterable, Function, int, boolean)
フィールド io.reactivex.internal.operators.PublisherZip.sources
ローカル変数名 sources
該当箇所 PublisherZip.java (line 40)

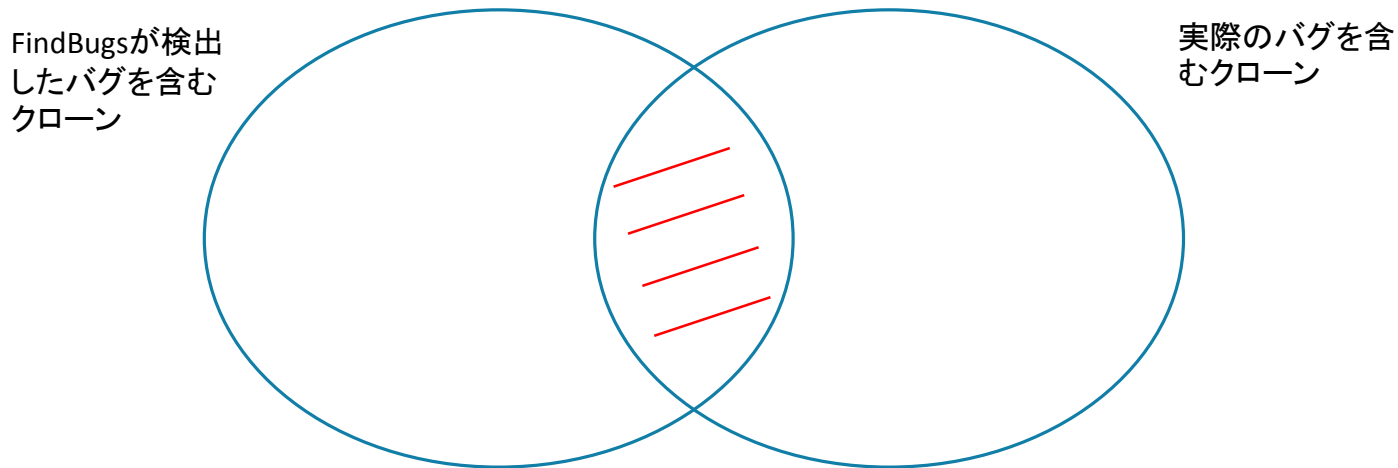
実際のバグを含むクローンの抽出

- 不具合修正コミットで修正された箇所がクローンに含まれていれば、クローン内部に実際のバグが含まれているとする



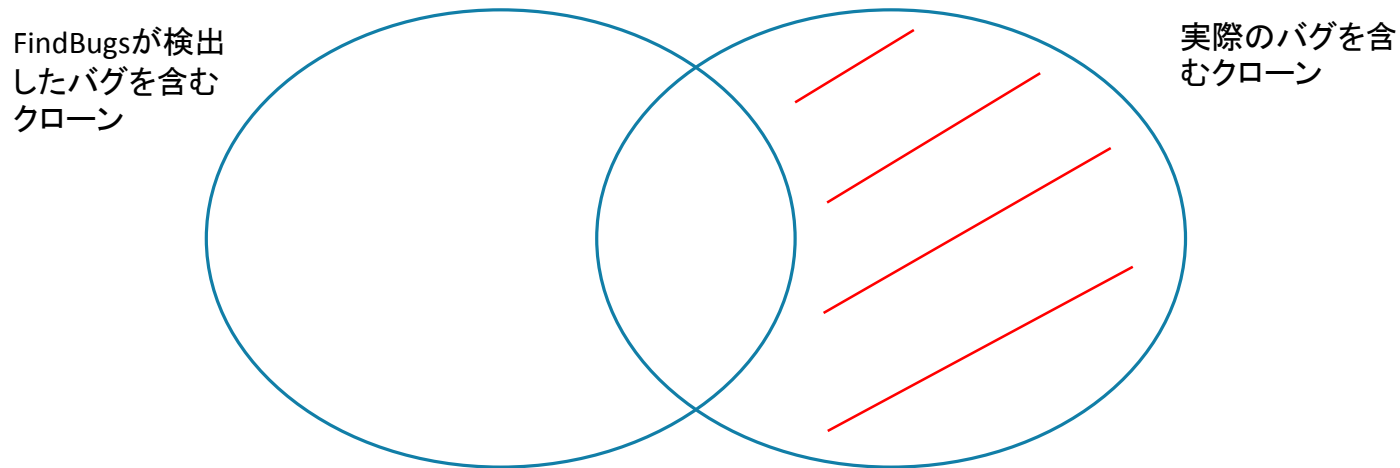
FindBugsのバグと実際のバグの両方を含むクローン

- 両方のバグを含むクローンを調べることで現在の静的解析ツールがクローンに含まれるバグをどれだけ検出することができるかを知ることができる



FindBugsでは検出できなかったバグを含むクローン

- このクローンの特徴を調べることでどのようなバグや危険性がクローンに含まれているかを明らかにすることができる



リサーチクエスト

- RQ1: FindBugsのバグと実際のバグの両方を含むクローンはどれだけ存在するか
- RQ2: FindBugsで検出できなかったバグを含むクローンのソースコードの特徴はどのようなものがあるか
- RQ3: FindBugsで検出できなかったバグを含むクローンのバグ票にはどのような特徴があるか

対象プロジェクト

- RxJava
- Javaでリアクティブプログラミングを行うためのライブラリ
- 対象期間
 - 2012年3月18日～2017年3月29日

RQ2とRQ3について

- RQ2はソースコードメトリクスを測ってどのような特徴があるかを調べる
- RQ3はバグ票を目視することでどのような特徴があるかを調べる

進捗状況

進捗状況

- 実際のバグを含むクローンを抽出するために、不具合修正コミットで修正されたクローンを収集した
 - 約5500個検出
- 現在RQ1とRQ2を調査中