



Commit Min

Jacob Ward: jward23, Evan Blajev: eblajev, Sarah Zhou: sarahsz,
Julius Christenson: juliusc, Conner Knight: connerk



Example of a Non-Minimized Commit

Say we have this program with a bug in it...

```
public static void main (string[] args) {  
    my_string = "hello";  
    for(int i = 0; i < my_string.length()-1; i++) {  
        System.out.println(my_string.charAt(i))  
    }  
}
```



Example of a Non-Minimized Commit

```
public static void main (string[] args) {  
    my_string_better_name = "hello";  
    for(int i = 0; i < my_string.length()-1; i++) {  
        System.out.println(my_string_better_name.charAt(i))  
    }  
}
```



Example of a Non-Minimized Commit

```
public static void main (string[] args) {  
    my_string_better_name = "hello";  
    //This should loop over the string?  
    for(int i = 0; i < my_string.length()-1; i++) {  
        System.out.println(my_string_better_name.charAt(i))  
    }  
}
```



Example of a Non-Minimized Commit

```
public static void main (string[] args) {  
    my_string_better_name = "hello";  
    //This should loop over the string?  
    for(int i = 0; i < my_string.length(); i++) {  
        System.out.println(my_string_better_name.charAt(i))  
    }  
}
```



Example of a Minimized Commit

```
public static void main (string[] args) {  
    my_string = "hello";  
    for(int i = 0; i < my_string.length(); i++) {  
        System.out.println(my_string.charAt(i))  
    }  
}
```



Problem and Motivation

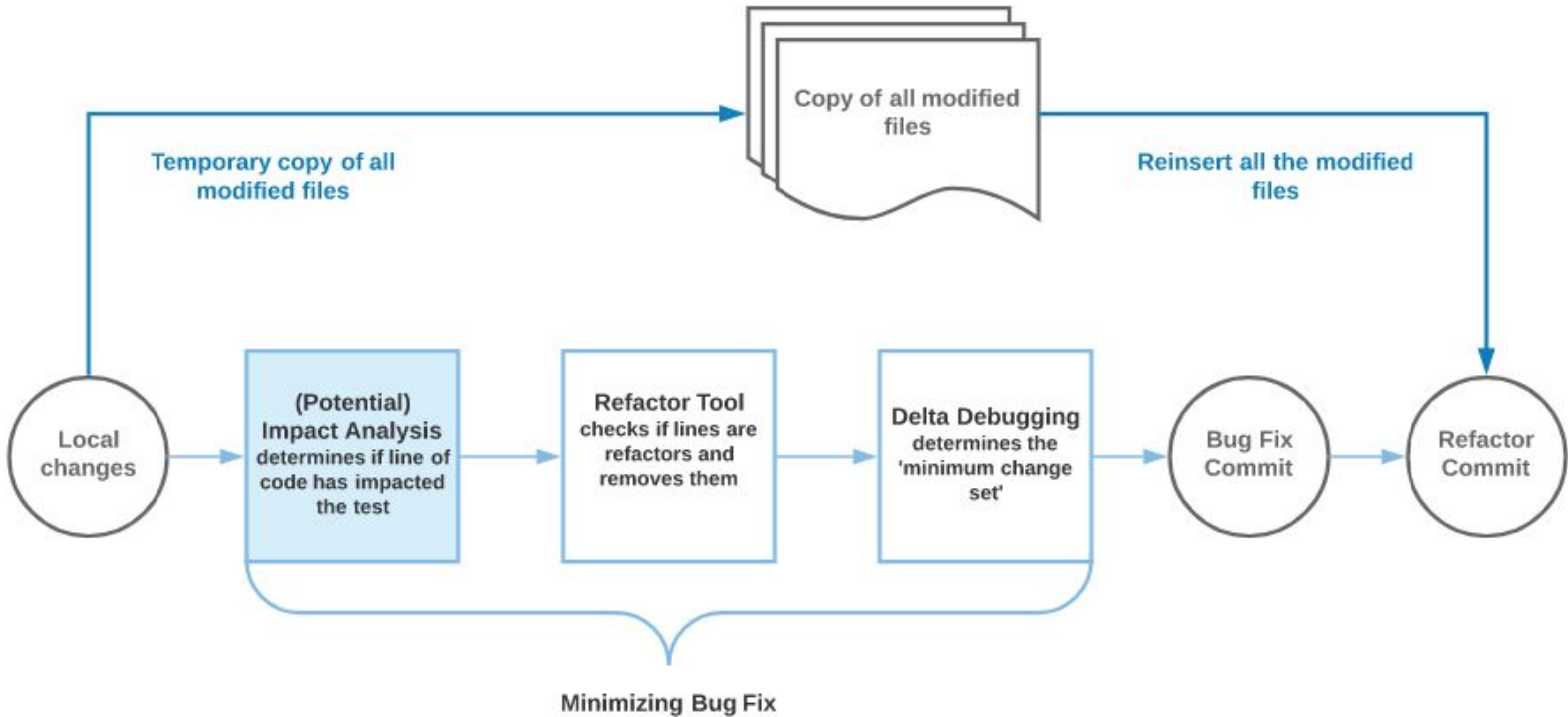
- Bug fixes often aren't isolated in commits
 - Refactorings and other code changes
- Problem when revisiting code or others try to understand fix
 - Developers waste valuable hours
- **Our Goal**
 - *Allow developers to push commits consisting of only the necessary changes for a bug fix*



Current Approach

- Git methods that require developer initiative
 - `git add -patch`
 - `git rebase -i`
- Requires user to manually select chunks of code to commit
 - User must already know bug fix
- Manually remove lines

Our Approach





Our Approach

- Main tool: Delta
 - Takes the lines of a file and determines what is used to pass the test
 - Breaks files into chunks of code [1-500][501-1000]
 - Check if test passes without those lines of code, repeats

Demonstration of Tool



```
Terminal - juliusc@localhost: ~/hellogitworld
File Edit View Terminal Tabs Help
[juliusc@localhost hellogitworld]$ echo "//this is a comment" >> src/main/java/com/github/App.java
[juliusc@localhost hellogitworld]$ echo "//this is a second comment" >> src/main/java/com/github/App.java
[juliusc@localhost hellogitworld]$ echo "this is a third comment" >> src/main/java/com/github/test.txt
[juliusc@localhost hellogitworld]$ git add .
[juliusc@localhost hellogitworld]$ git commit -m "This is a test commit"
Welcome to Commit-min, do you want to minimize your commit? Pass in flags if needed (Y/N)
y
Starting Commit-min
Please enter the name of the test to run Commit-min with (TestFileName#testName)
AppTest#testApp
```



Demonstration of Tool

```
diff --git a/src/main/java/com/github/App.java b/src/main/java/com/github/App.java
index 53efb19..8d9a7f2 100644
--- a/src/main/java/com/github/App.java
+++ b/src/main/java/com/github/App.java
@@ -9,3 +9,5 @@ public class App {
    }
}

this line has been removed from the commit: //this is a comment
this line has been removed from the commit: //this is a second comment
diff --git a/src/main/java/com/github/test.txt b/src/main/java/com/github/test.txt
new file mode 100644
index 0000000..1e64830
--- /dev/null
+++ b/src/main/java/com/github/test.txt
@@ -0,0 +1 @@
this line has been removed from the commit: this is a third comment
```



Results (Or Lack Of)

- Tested on defects4j (real world commits minimized) to attempt to minimize a commit and then compare against the human minimized defects4j commit
- Results to compare: How many lines we got rid of that defects4j humans left in, and how many lines we left in that defects4j humans got rid of



Results (Or Lack Of)

- While our tool can work on smaller commits, with larger commits, the exponential runtime with respect to lines in the file leads to the program not terminating with large “real world” commits



Experiments

- Through our testing, we found the main problem:
 - Length of time to minimize commits
 - Real files are often 1000s of LOCs, and we scan every line of code to determine what is used for the test.
- Tool ran for 3 hours
 - Reduced a file for one commit from ~700 lines to ~400 lines, did not finish



What to do now?

- Designing a solution that runs efficiently
 - *Potential Fix*: Instead of running delta on the entire file, run only on the diffs
 - *Potential Fix*: Decrease the granularity of delta debugging tool, decreasing accuracy also decreasing runtime (run on hunks of diff, not lines of files)