Looking for Patterns: Characterizations of 'If' Conditionals in Buggy Code

Megan Kierstead
Wellesley College

Advisor: Professor Jim Whitehead
University of California – Santa Cruz

Motivation

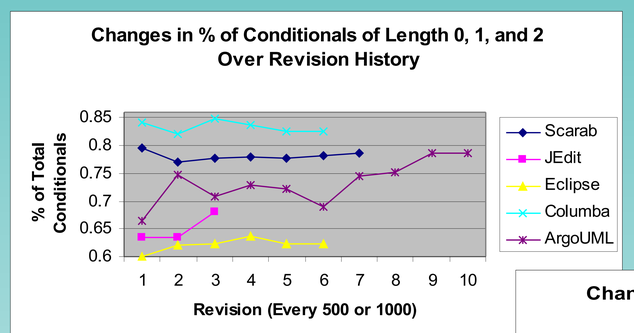
To find and characterize patterns in the length and distribution of conditionals in buggy code using databases of bug memory fixes.

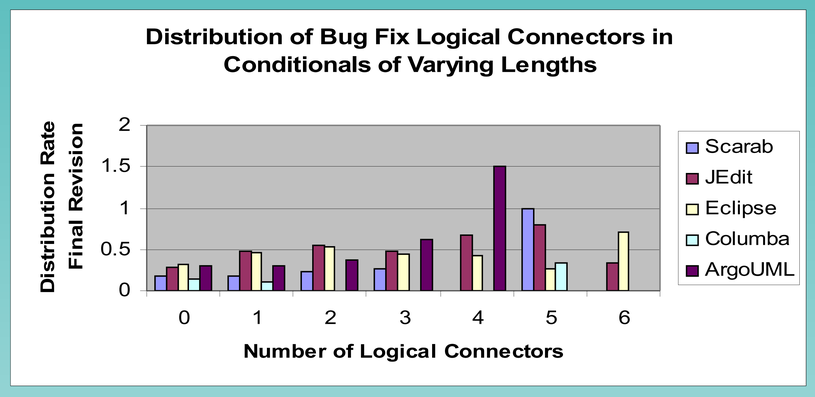
- Conditionals become logically complex as the length increases and might be difficult for programmers to code correctly.
- Conditionals are an important feature of programming languages and characterizations of 'if' bugs could lead to better language design or more conscientious programming.
- Within 5 Java open source projects, 8.0%-17.0% of all bug fix patterns involve the language keyword 'if'.

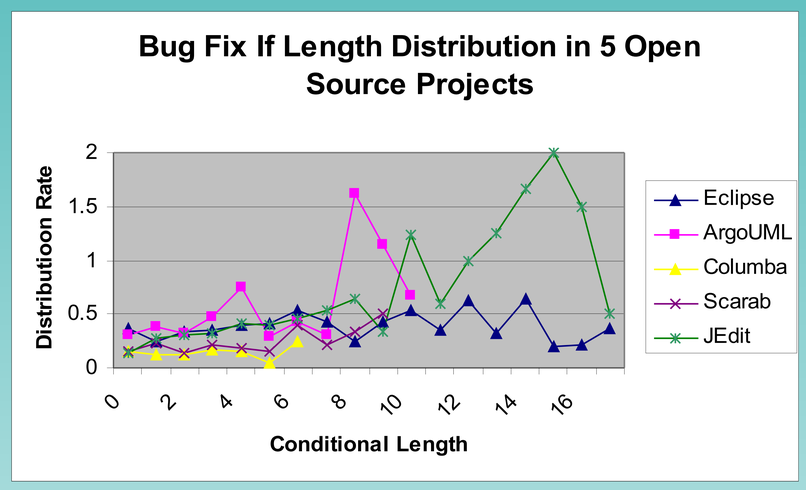
Experiments

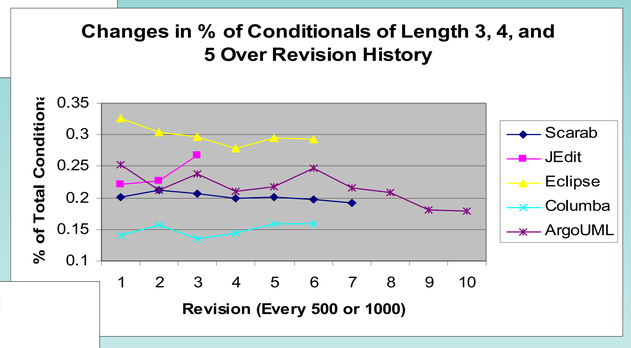
Using databases created by graduate students Kai Pan and Sunghun Kim, I extracted the following characterizations:

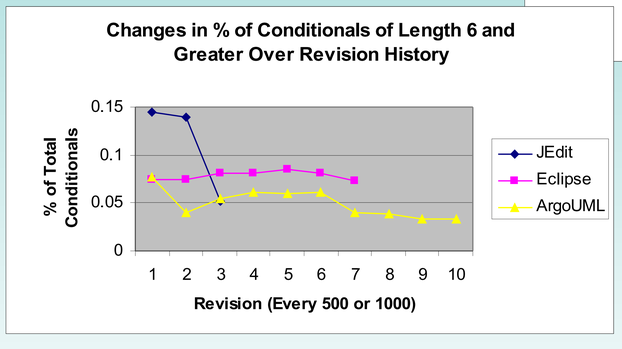
- Number of logical connectors (&&, ||) per conditional.
- Number of variables and operators per conditional.
- Changes in the length of conditionals over program revision history.
- Comparisons of buggy if conditionals to overall program if conditionals.
- Changes in length of conditionals compared to program length.
- The distribution rate of bugs, which is the total number of bugs found divided by the number of bugs found in the final revision











Results

While there are some interesting patterns among if conditionals, the results I extracted did not definitively say whether or not code becomes more buggy as conditional length increases.

- A length 39 conditional from Eclipse:

```
if (((splitOperatorsCount == 2 && splitOperators[1] == TokenNameDOT && splitTokenDepth
== 0 && lastOpenParenthesisPosition > -1) || (splitOperatorsCount > 2 && splitOperators[1]
== TokenNameDOT && splitTokenDepth == 0 && lastOpenParenthesisPosition > -1 &&
lastOpenParenthesisPosition <= options.maxLineLength) ||
(separateFirstArgumentOn(firstTokenOnLine) && splitTokenDepth > 0 &&
lastOpenParenthesisPosition > -1)) && (lastOpenParenthesisPosition <
splitScanner.source.length && splitScanner.source[lastOpenParenthesisPosition != ')']));
```

This work was completed as part of UCSC's SURF-IT summer undergraduate research program, an NSF CISE REU Site. This material is based upon work supported by the National Science Foundation under Grant No. CCF-0552688.