Review 5

2/1/12

CS6V81.502

Seungtack Baek

**Finding What’s Not There:**

**A New Approach to Revealing Neglected Conditions in Software**

As the authors indicate in the paper, forgetting a condition is a major cause of a bug in software. A developer might not think thoroughly (or simply does not know the specification well enough, whatever the reason being), so one can forget to include one or more condition in the if-statement. Even worse, sometimes whole if-statement is dropped, such as checking for nullity. This type of logical error can be fatal to the software concerning functionality and security. So in this paper, the authors suggest code inspection, boundary value testing, special value testing, mutation testing and model checking for possible alternatives for detecting neglected conditions. Yet, in this study, they employed the idea of dependency graph to mine the pattern so they could construct programming rule. In turn, this rule was used to determine neglected conditions.

The result, however, is very weak in the sense that this study only considered one source code. Also, in their conclusion, the authors vaguely say that their result “suggest” the feasibility of rule discovery. Their motivations for finding rules via searching for patterns were great, and the idea of using frequent subgraphset mining graph was great (which I think gives more information than frequent itemset mining – is this correct?) Yet, their paper was really hard to read. The introduction was little bit lengthy, which could be broken into smaller sections and they assumed too much on the graph theory knowledge of the readers.

The other regrettable aspect of the paper is that, while I was expected to see more on detecting neglected conditions, I felt that the authors spent much more time on data mining itself. I wish they said more about the neglected conditions itself. (or simply change the title)

.

**Question:**

1. This paper seems utterly relatable to the previous reading on PR Miner. Did you choose this on purpose?
2. Did you find reading this article to be hard? (say, what percent of the paper did you get? – because I only got 30% out of it. It just assumes that the reader has substantial knowledge on graph theory)