Review 23

4/17/12

CS6V81.502

Seungtack Baek

**A Study of the Uniqueness of Source Code**

In this paper, Gabel et al. studied uniqueness of source code at “various granularity.” Even if there are similar tools exists, such as code clone detection, code reuse, etc. study that measures degree of uniqueness, in terms of syntactic redundancy, is never studied.

They start their study with a hypothetical question: “where you have to build a project solely from copying-and-pasting existing source code, how much can you finish?” Even if this question is almost impossible to answer (how do you know how big should a project be?), they tried to solve this question by the following setup:

1. Two sets of target projects: 30 SourceForge projects under full precision (depth) and 6,000 corpus projects under stochastic estimation (breadth).

2. Two levels of abstraction: none and renamed identifiers.

3. Exact ‘matches’ and matches allowing for maximum Hamming distances of 1, 2, 3, and 4.

4. As many levels of granularity as our computing resources will allow.

They will proceed to the evaluation in the next half of the study.

The main question I had for this study was this: why would we even care about uniqueness of the source? From my experience, if I do not know how to do certain things, say reading a file into a string, I just google how to do that and modify the sample code to make it suitable for my project. Can you call this unique..? well..

Yet, soon I figured that there are three (according to this paper) usages for the uniqueness. Of those potential usages, I found code completion tool to be most attractive. If a tool can complete a function for me (and the right one, of course), I will spend some money for the tool. So this tool may be marketable in the future.

**Question**

1. What do you mean by “four months of CPU time,” in section 1?
2. Would using a different framework change the result of syntactic