Review 24

4/19/12

CS6V81.502

Seungtack Baek

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

In the second half of this paper, Gabel et al. studied uniqueness of source code at “various granularity.” In this section, the researchers actually used the tool to see how much of the code is syntactically redundant –in other words, unique.

To carry out the experiments, they took 30 projects (10 projects each from C, C++ and Java projects) as subject projects. For each project, the researchers recorded all syntactic redundancy for variable parameter settings such as g-value, minimum hamming distance etc.

The result was quite interesting. Most of the projects, for high g-values, were found to be pretty unique. Yet, the charts on page 7 can be misleading. The biggest danger in statistics comes when you eyeball a graphical representation of a distribution (since the charts/graphs can be easily distorted in favor of author’s opinion). So it is better idea to include statistical figures along with the charts, such as p-value (which I do not think that I found a value for it).

Going back to the result, I do agree with the result that they obtained. Even if there could be a lot of “exact” matches between single lines, if you increase the scope to 5-6 lines, it will be hard to find exact matches (or even with 3-4 hamming distances – which can allow function call changes as well).

However, I like how they specify the machines they used, such as specifying CPU/RAM they used. This gives us more clear idea on the performance of the tool. Most of the papers we read so far did not bother to include the specification of their target machine, which leaves a lot of room for questions.

**Question**

1. What is g value? (more specifically, why did you choose such values? Since 77 looks like there is reasoning behind it…)
2. Although I agree with you on that “only a small fraction of code in high-level languages is likely to be truly platform-specific,” can you elaborate more on that?