Validation

Scott Kuroda

March 13, 2012

My thesis topic is project based. As seen in my overall presentation, the actual topic still needs some refinement. However, the framework for validation is similar across several different components of the system. However, depending on the route taken, there may be components of experimentation as well as standard project based evaluation.

Beginning with a high level evaluation, the first step would simply be to see if I, to some extent, reached all of my defined requirements. Of course, this is more of a qualitative analysis, but would serve as an initial step before diving into the details of more quantitative validation. It is also something that would apply to any portion of the project that I choose as my thesis topic. Next I will discuss validation methods for various components.

Of particular interest to me is working with the existing static analysis tools. Doing this for my thesis would require developing a framework in which various static code analysis tools could be condensed into a single report. To validate this work would require several different components. First, I should be able to take an arbitrary code analysis report, and incorporate it into the report generation system. Even if my focus is on the framework, rather than the incorporation of a wide variety of static code analysis tools, I should still be able to take a report, modify in some way similar to that of a parser, and incorporate it into a final report. In addition, I should be able to take multiple arbitrary reports, and combine them into a single report, without losing any reported warnings or errors. This would require running a series of previously generated reports to be run through the framework. These reports would then have to be compared, by hand, to ensure that they are complete. It may also be valuable to show the final generated reports to a third party, to get additional feedback on the formatting as well as content of the final reports. Of course, this may be a more qualitative rather than quantitative analysis. Because the results may also be passed to another component of the system, it would also be important to validate a more symbolic output. In terms of content, this should match the information being passed back to the end users, but in a more format or object that is easily parsable.

Another area of interest is the rule generation. This particular component requires the generation of rules from both static analysis methods as well as human feedback. The first specific set of analysis done would be comparing the inputs to the rules generated. In addition, because this system would ideally learn, there would also need to be further an additional set of comparisons after the system is trained to some extent. This output would then need to be compared to the original output, to see how well the system learned. As with the static report generation, there will also be a component requiring analysis from a third party. Because the rules would be distributed to an end user, as well as internal to the system, it is important that the generated results are in a human readable format. Again, as with the static analysis results, there would also need to be validation of some easily parsable object, to be passed along to other programmatic modules in the system.

No matter what the main focus of my thesis is, there many of the concepts in validation are similar. For each, there will be some analysis by a third party. Though in my mind this is more of a qualitative analysis, given a large enough sample size, there may be some type of statistical analysis that can be done based on the responses from the third party individuals. In addition, there are elements of analysis regarding the outputs generated, be they reports or rules, based on the sets of inputs. The rule based generation would also include analysis over time as the system should be learning as more cases are submitted.