Title: Soundness versus precision
Date: 02.02.2014
Category: Static and dynamic analysis
Tags: pelican, publishing
Author: Ilija Radosavovic
Summary: Comparison between static and dynamic analysis

Static and Dynamic analysis developed side by side in different communities and have been regarded as distinct disciplines. Furthermore, the domains in which one of them shines represent the nightmare for the other.

Static Analysis inspects program code, and strives to conclude all possible outcomes of program execution. The most common static analyses are compiler optimisations. Static analysis is sound and conservative, which guarantees the correctness of its conclusions. In other words, "Static analysis never makes mistakes". Its conservatism reflects in making weaker, easier to establish, statements that are guaranteed to be true and thus preserve soundness. Therefore, the analysis output may not be precise enough to be useful. For instance, for some function g, the statement "g returns a number" is weaker than the statement "g returns an even number".

In contrast, conclusions of dynamic analysis are based on executing a program, and observing its behaviour during the runtime. Testing and profiling are standard dynamic analysis. Dynamic analysis is precise and exact because it does not require any abstractions or estimations. It deterministically answers questions about computed values, memory consumption or the length of the execution. Dynamic Analysis is quick to perform, and can take as little time as program execution.

Although, some static analyses are performed quickly, generally, obtaining precise results demands many complicated and time consuming computations. Furthermore, certain problems, such as pointers analysis, are almost unsolvable. Conversely, the same problem is solved at runtime, using a single machine cycle to compare the pointers.

The results of dynamic analysis cannot be generalised to future executions. Furthermore, there is no guarantee that the set of test cases, over which the program was run, covers all possible executions of the program. One of the main challenges of dynamic analysis is constructing a test suit, which covers all representative cases, and thus reveals properties of the program. Despite the fact that unsound dynamic analysis often faces criticism, it is used on many occasions because it delivers a sufficient quantity of valuable information.

Since both of the analyses collect different information, performing one analysis, then the other is more powerful than performing either one on its own. Static and dynamic analysis can supplement each other by providing, otherwise unavailable, information. For instance, static analysis can reduce the collection of

data by guaranteeing that a smaller amount of information is sufficient, which makes dynamic analysis quicker and more efficient.

Traditionally, static and dynamic analysis have been viewed as distinct approaches with different techniques. However, static and dynamic analysis can interact by enhancing each other, and blending the strengths of both approaches.

Bibliography

Ernest, M. D. (2003). *Static and dynamic analysis: synergy and duality.* Portland, OR, USA: WODA 2003: ICSE Workshop on Dynamic Analysis.