The determination of semantic equivalence is an undecidable problem; however, this dissertation shows that a reasonable approximation can be obtained using a combination of static and dynamic analysis. This study investigates the detection of functional duplicates, referred to as semantic method clones (SMCs), in Java code. My algorithm extends the input-output notion of observable behavior, used in related work [1, 2], to include the effects of the method. The latter property refers to the persistent changes to the heap, brought about by the execution of the method. To differentiate this from the typical input-output behavior used by other researchers, I have coined the term method IOE-Behavior which means its input-output and effects behavior. Two methods are defined as semantic method clones, if they have identical IOE-Behavior—that is, for the same inputs (actual parameters and initial heap state), they produce the same output (that is result— for non-void methods, and final heap state).

The detection process consists of two static pre-filters used to identify candidate clone sets. This is followed by dynamic tests that actually run the candidate methods, to determine semantic equivalence. The first filter groups the methods by type. The second filter refines the output of the first, grouping methods by their effects. This algorithm is implemented in my tool JSCTracker, used to automate the SMC detection process.

The algorithm and tool are validated using a case study comprising of 12 open source Java projects, from different application domains and ranging in size from 2 KLOC (thousand lines of code) to 300 KLOC. The objectives of the case study are posed as 4 research questions: can method IOE-Behavior be used in SMC detection? What is the impact of the use of the pre-filters on the efficiency of the algorithm? How does the performance of method IOE-Behavior compare to using only input-output for identifying SMCs? How reliable are the results obtained when method IOE-Behavior is used in SMC detection? Responses to these questions are obtained by checking each software sample with JSCTracker and analyzing the results.

The number of SMCs detected range from 0 - 45 with an average execution time of 8.5 seconds. The use of the two pre-filters reduces the number of methods that reach the dynamic test phase, by an average of 34%. The IOE-Behavior approach takes an average of 0.010 seconds per method while the input-output approach takes an average of 0.015 seconds. The former also identifies an average of 32% false positives, while the SMCs identified using input-output, have an average of 92% false positives. In terms of reliability, the IOE-Behavior method produces results with precision values of an average of 68% and recall value of 76% on average.

These reliability values represent an improvement of over 37% (for precision) and 30% (for recall) of the values in related work [3, 4]. Thus, it is my conclusion that IOE-Behavior can be used to detect SMCs in Java code with reasonable reliability.

Major: Computer Science

Educational Career:
Bachelor's of Computer Science (Minor in Education), BS, 2002, University of the West Indies, St. Augustine, Trinidad
Master's of Computer Science, MS, 2005, University of Central Florida

Committee in Charge:
Gary T. Leavens, Chair, Computer Science
Ali Orooji, Computer Science
Charles E. Hughes, Computer Science