Masters Final Oral Defense

Monday, April 10th, 2017
223 Atanasoff Hall at 2:00 p.m.

Ashwin Kallingal Joshy

Computing Change of Invariants to Support Software Evolution

Software is always evolving. In the recent years the development community has shifted towards Agile development paradigm resulting in faster release cycle. This emphasis on speed is, generally, accompanied by an increase in the number of bugs and reduced focus on updating non-functional software artifacts like specification document. Recent studies have found that developers find it difficult to determine whether a change might break code elsewhere in the program, resulting in 25% of bugs fixes to be incorrect or buggy. A method to capture the semantic changes between different versions of a program is vital in understanding the impact of the change and in preventing bugs.

An invariant is a condition that is always true at a given program point. Invariants are used to specify the requirements and desired behavior of a program at any program point. The difference in invariants between different program versions can be used to capture the changes made to the program. In this thesis, we use the change of invariants as a way to capture the semantic changes over different program versions. We designed a static demand-driven algorithm for automatically computing the change of invariants between different versions of a program. To evaluate the algorithm and its ability to capture semantic changes over different program versions, a prototype framework called Hydrogen was build. Our experimental results show that Hydrogen is able to compute the change of invariants between different versions of the programs in the benchmark, and the computed change of invariants can be used for understanding the changes and generating assertions to prevent similar bugs in future.

Major Professors:

Wei Le

Committee Members:

Samik Basu

Jin Tian

IOWA STATE UNIVERSITY

Department of Computer Science

www.cs.iastate.edu