NATIONAL UNIVERSITY OF SINGAPORE

School of Computing

PH.D DEFENCE - PUBLIC SEMINAR

Title: **Enhancing Total Correctness Proofs in Program Verification**

Speaker: Mr Le Ton Chanh

Date/Time: 6 March 2015, Friday, 03:00 PM to 04:30 PM

Venue: Executive Classroom, COM2-04-02

Supervisor : Dr Chin Wei Ngan, Associate Professor, School of Computing

Abstract:

Proving the total correctness of large-scale software systems with complex safety and liveness properties is a great challenge in program verification. To specify these properties and verify or analyze them successfully, the software verification systems usually require expressive specification logics with scalable verification techniques to be developed. However, recent advances in software verification mainly focus on partial correctness with safety properties. The aim of this thesis is to develop methodologies to enhance expressiveness, focusing on program termination and non-termination reasoning, and scalability, focusing on the concept of modularity, of total correctness proofs in program verification.

Firstly, we propose a logical framework for specifying and verifying termination and non-termination properties of programs. These properties are defined as resource capacity of execution length and reasoned about in terms of resource reasoning. This approach allows the termination and non-termination assertions to be seamlessly integrated into available logics for functional properties to conduct more intricate termination and non-termination proofs. Its result is a unified framework, which can verify both partial correctness, termination and non-termination of various programs, including heap-manipulating programs. Experimental evaluation shows the expressiveness, usability and practicality of our approach on over 300 challenging programs.

Secondly, we propose a modular inference mechanism for summarizing termination and non-termination behaviors of each method in programs. We extend the proposed termination logic with second-order termination predicates and leverage the available Hoare-style verification

infrastructure to collect a set of relational assumptions on them.
We then solve these assumptions with case analysis to determine both termination and non-termination behaviors of analyzed methods.
The inference result is expressed in a compatible logic form of the underlying verification system, so that they can be re-verified.
Experimental evaluation on the benchmark suite of a recent termination competition shows the scalability and efficiency of our mechanism against state-of-the-art termination analyzers.

Lastly, we propose a formal framework for proof slicing in verification that can aggressively reduce the size of the discharged proof obligations as a means of performance improvement. Our proposal is built on top of existing automated theorem provers and can be viewed as a re-engineering effort in proof decomposition that attempts to avoid large-sized proofs for which these provers may be particularly inefficient. Our theoretical development is supported by experimental results, which show significant improvements in the verification of complex programs.