

**DOCTORAL CANDIDATE:** THI MAI THUONG TRAN  
**DEGREE:** Philosophiae Doctor  
**FACULTY:** The Faculty of Mathematics and Natural Sciences  
**DEPARTMENT:** Department of Informatics  
**AREA OF EXPERTISE:** Computer Science  
**SUPERVISORS:** Martin Steffen, Olaf Owe, Einar Broch Johnsen  
**DATE OF DISPUTATION:** 11th of June 2013

**DISSERTATION TITLE:** *Compositional Formal Analysis for Concurrent Object-Oriented Languages*

Concurrency is a ubiquitous phenomenon in modern software ranging from distributed systems communicating over the Internet to communicating processes running on multi-core processors and multi-processors. Therefore modern programming languages offer ways to program concurrency effectively. Still, writing correct concurrent programs is notoriously difficult because of the complexity of possible interactions between concurrent processes and because concurrency-related errors are often subtle and hard to reproduce, especially for safety-critical applications. This thesis develops and formally investigates different static analysis methods for various concurrency-related problems in concurrent object-oriented programming languages to guarantee the absence of common concurrency-related errors, hence contribute to the quality of concurrent programs.

Aspects covered by our analyses involve lock-based concurrency, transaction-based concurrency, resource consumption and inheritance. In the lock-based setting, using explicit locks in a non-lexical scope to protect critical regions might be the source of aliasing problems or misuse of locks. In the transaction-based model, a similar problem of misuse of non-lexical transactions can happen in the same way as the misuse of explicit locks. Furthermore, for the purpose of checking conflicts and supporting rollback mechanisms, additional storage are required to keep track of changes during transactions' execution which can lead to resource consumption problems. So it is useful to investigate different analysis methods to achieve safe concurrent programs. In open systems, the combination of inheritance and late-binding problematic, e.g., replacing one base class by another, seemingly satisfying the same interface description, may break the code of the client of the base class. This thesis also investigates an observable interface behavior of open systems for a concurrent object-oriented language with single-class inheritance where code from the environment can be inherited to the component and vice versa.

All analyses developed in this work are formulated as static type and effect systems resp. an open semantics based on a type and effect system. As usual for type and effect systems, they are formalized as derivation systems over the syntax of the languages, and thus compositional. In all cases, we prove the correctness of the analyses. When based on a rigorous formal foundation, the analyses can give robust guarantees concerning the safety of the program. To tackle the complexity of large and distributed applications, we have insisted that all the analysis methods in this thesis should be compositional. In other words, our analyses are scalable, i.e., a composed program should be analyzed in a divide-and-conquer manner relying on the results of the analysis of its sub-parts, so they are useful in practice.