

PhD FINAL ORAL EXAMINATION

Thursday, April 14th
10:00a.m. @ 223 Atanasoff

Yuheng Long

Faculty Advisor: Professor Hridesh Rajan

Formal Foundations for Hybrid Effect Analysis

Type-and-effect systems are a powerful tool for program construction and verification. Type-and-effect systems are useful because they can help reduce bugs in computer programs, enable compiler optimizations and also provide a kind of program documentation. As software systems increasingly embrace dynamic features and complex modes of compilation, static effect systems have to reconcile over competing goals such as precision, soundness, modularity, and programmer productivity. In this thesis, we propose the idea of combining the static and dynamic analysis for effect systems to improve precision and flexibility.

We describe intensional effect polymorphism, a new foundation for effect systems that integrates static and dynamic effect checking. Our system allows the effect of polymorphic code to be intensionally inspected. It supports a highly precise notion of effect polymorphism through a lightweight notion of dynamic typing. When coupled with parametric polymorphism, the powerful system utilizes runtime information to enable precise effect reasoning, while at the same time retains strong type safety guarantees. The technical innovations of our design include a relational notion of effect checking, the use of bounded existential types to capture the subtle interactions between static typing and dynamic typing, and a differential alignment strategy to achieve efficiency in dynamic typing. We also introduce a new kind of effect system, where the computational effect of an expression can be programmatically reflected, passed around as values, and analyzed at run time. A broad range of designs “hard-coded” in existing effect-guided analyses can be supported through intuitive programming abstractions.

Finally, we show the potential benefit of intensional effects by applying it to an event-driven system to obtain safe concurrency. The technical innovations of our system include a novel effect system to soundly approximate the dynamism introduced by runtime handlers registration, a static analysis to pre-compute the effects and a dynamic analysis that uses the precomputed effects to improve concurrency. Our design simplifies modular concurrency reasoning and avoids concurrency hazards.

IOWA STATE UNIVERSITY
Department of Computer Science