STEVENS
INSTITUTE OF TECHNOLOGY
1870

Ph.D. DISSERTATION DEFENSE

Candidate:	Adam Chen
Degree:	Doctor of Philosophy
School/Department:	Charles V. Schaefer, Jr. School of Engineering and Science / Computer Science
Date:	Thursday 4 December 2025
Time/Location:	11:00AM – 12:00PM EST; Remote: https://stevens.zoom.us/j/94062277271
Title:	Commutativity Analysis for Automatic Parallelization
Chairperson:	Eric Koskinen, Department of Computer Science, Charles V. Schaefer, Jr. School of Engineering and Science
Committee Members:	Arthur Azevedo de Amorim, Department of Computer Science, Golisano College of Computing and Information Sciences (Rochester Institute of Technology) Eduardo Bonnelli, Department of Computer Science, Charles V. Schaefer, Jr. School of Engineering and Science Tegan Brennan, Department of Computer Science, Charles V. Schaefer, Jr. School of Engineering and Science

ABSTRACT

Parallelization of code is increasingly important to high performance. One such enabler of parallelization is commutativity. However, code fragments may not always commute. One approach is to use *conditional commutativity*, where code fragments commute under some precondition called the *commutativity condition*.

This dissertation introduces new methods for automatic parallelization of programs by exploiting conditional commutativity. I introduce a method for automatic parallelization of programs with commutativity conditions via the Veracity programming language. Sequential and parallel semantics of so-called `commute` blocks are given. Safe parallelization requires scoped serializability, a novel correctness condition that is a strengthening of serializability. Next, in the context of a whole-program parallelization in Veracity2G, I describe a novel runtime model and implementation for concurrent execution of programs with commutativity conditions across non-neighboring code regions.

These techniques naturally depend on knowing commutativity conditions, which must be discovered and verified. This dissertation thus also improves automation of commutativity condition verification and synthesis in two ways. First, I present a new tool Servois2: a generational improvement on an existing tool for synthesis of commutativity conditions of abstract data types (ADTs). The tool reasons about a larger number of theories and outperforms its predecessor. Servois2 is used as the commutativity condition reasoning backend for Veracity and Veracity2G. Second, a novel abstract interpretation framework for extending a language with

ADTs is presented. Formalization of *sufficient commutativity abstractions* is given, and the ability to statically guarantee these abstractions is shown.

In sum, the work presented explores the process of using conditional commutativity for performance benefit, starting from obtaining the conditions through the parallelization of code.