

Since the use of a computer to prove the four-color theorem in 1976, computers have only gained popularity among mathematicians. While there are still reasons to dislike the original proof as lacking any insight into the problem, it stands that the computer aided in solving the question. While there are many different ways a mathematician can use a computer, I see the need for developing our tools in making this interaction as seamless as possible. Email has allowed us to communicate more quickly and cut down response times, mathematical programming languages like Maple and Mathematica have allowed us to run previously difficult computations in seconds, online databases such as the Online Encyclopedia of Integer Sequences has allowed us to search for sequences across all literature, and open sources books such as the Stacks Project allow us to effectively organize and disseminate research in more meaningful ways. We can all agree computers have affected mathematics, and the better tools we have available, the more progress within mathematics we can make.

This feeling that computers are having a revolutionary affect on mathematics has driven me towards two main fields; automated theorem proving, and formal verification. It has also taken me down side streets towards areas like homotopy type theory, constructive mathematics, and category theory where we see ideas mostly about formal verification being used. I first learned about formal verification after being told about the Coq proof assistant where I then read about the other languages available such as LEAN, Isabelle and Agda. Once I understood some of the basic ideas of formal verification it was a relatively small leap to think about how we can harness this power of formal verification to automate certain extensions/generalizations of theorems, lemmas, etc. I fantasized about building databases of formally proven theorems that could be indexed and searched by all types of mathematicians from around the world. Need a lemma for your big theorem? No worries, just search a few keywords and see if someone else has proven it, possibly 15 years ago in a different field in an obscure paper and you'll never know about it. I understand this is a pipe dream, but I don't believe it's impossible in the long term future. Much of the work done in this field will have lasting impacts for years to come and it's important we look ahead at what may be needed.

Carnegie Mellon has a strong connection with this field, and has even been largely connected with the development of the LEAN proof assistant. With multiple researchers who study directly in this field, or in adjacent ones, CMU is an ideal place to study, and begin research in formal verification and automated theorem proving. Along with having the requisite researchers, CMU's Pure and Applied Logic PhD program is perfect for the interdisciplinary research that is needed in this field. Mathematics, and computer science is clearly foundational to this work, but the philosophy behind using computers within mathematics is arguably some of the most interesting. Since Euclid's time, mathematics was something to be done by humans, and as electronics progressed calculators did the grunt work. We are now at a point where that may be changing. What does this mean for mathematics as a subject? Is proving a theorem now grunt work? What do mathematicians do now? Do we need to understand it if the computer does? There are lots of philosophical questions that are now opened because of this alliance. With a program that is designed for all three of these fields to come together, I see CMU to be one of the best, if not the best place to study formal verification and automated theorem proving.