Personal Statement

Intellectual Merit As a college freshman, I wanted to be a philosopher. I was enchanted by the field's combination of rigor and creativity and the frequently excellent socratic teachers I found there: for the first time, the answers were always just out of reach, leaving me empowered to do the hard exploration and analysis and reap its intellectual rewards.

It was another great teacher who showed me that the greatest demand for rigor and the most spellbinding creativity are to be found elsewhere. Early in my second year, I was conned into taking a graduate-level logic seminar. In over my head—we covered the first fourteen chapters of Jech's *Set Theory*—I nonetheless liked the water: here was something relentlessly creative, astoundingly deep, and completely precise. I haven't looked back since.

In the intervening years, I have acquired a strong background in mathematics and computer science. In the interest of equiping myself with as broad and powerful a set of tools as possible (and also because it's a great deal of fun), I have successfully tackled a significant amount of graduate coursework in mathematics, last year in complex analysis and this year in the beginnings of algebraic topology and smooth manifolds. Additionally, I have a growing record of graduate-level work in theory of computation: by the end of the year I will have taken graduate courses in programming languages, algorithms, and randomized algorithms.

I have continued, also, to avail myself of all the logic courses available to me. At UW, logic is taught in the philosophy department with strange and unrepresentative course numbering, but I have experience in set theory through the independence of CH from ZF, model theory through quantifier elimination and strong minimality, modal logic through standard completeness proofs, and proof theory through the incompleteness theorem. Where courses have been unavailable or too slow to be worthwhile, I have self-directedly read the material. This is the source of much of my logic and complexity backgrounds, from the standard works by Mates, Jech, Kunen, and Soare in logic and Sipser and Arora-Barak in complexity.

So we come to theory of computing. In logic, many of the most central results were proved seventy-five years ago. But here, while in many ways the questions are no less fundamental—Can fast randomized computations be derandomized efficiently? How can we extract enough information from computational structures to prevent our proofs from relativizing or naturalizing?—there is the feeling that many of the central results have yet even to be formulated. I find this immensely exciting. Furthermore, though I have just begun to learn about it, logic is finding a new interplay with the theory of programming languages, opening the door for investigation of new foundational questions at their interface. My growing research experience—on projects in complexity with Eric Allender and Paul Beame and on the design of the Dart programming langauge at Google—has only fed my excitement.

For all these reasons, I am interested in problems on both sides of the traditional "Theory A" and "Theory B" divide (named for the volumes of the *Handbook of Theoretical Computer Science*). My undergraduate education has avoided premature specialization, preferring breadth across mathematics, logic, and computer science with the understanding that the key to a new illumination of an open problem is often the ability to consider it in the light of intuitions from another field entirely. When the time comes to pick a thesis problem, I will of course have to narrow my focus, but I maintain that success in theoretical computer science, especially in areas where current techniques make progress unfeasible, demands both a deep and wide mathematical arsenal. I would love to go to CMU, where I could work with Jeremy Avigad or Ryan O'Donnell on logical and analytic techniques in complexity, respectively, or with Robert Harper at the intersection of algebra, programming languages, and proof verification.

Broader Impacts I experienced first-hand and have witnessed many times over the broad failures of our K-12 system to serve gifted students well, especially in mathematics and computer science: for every gifted child who is challenged by her teachers there are many others for whom school is a mindless chore. It is extremely important to provide opportunities to save such students from the tyranny of "plug-n-chug" mathematics. I am committed to organizing and participating in instruction and curriculum development for K-12 outreach programs in mathematics and computer science, especially those aimed at serving underserved gifted students. I will discuss my record and future plans in instruction and outreach.

Over the course of my undergraduate career, I have established a strong record of teaching and tutoring and a growing record of K-12 outreach. In the Philosophy department I was for two years a writing tutor, working one-on-one with students to focus their philosophical understanding and clarify their writing. Though my teaching has since seen a change of subject matter, the skills—finding the right questions to ask, assessing on-the-fly what might grab a student's attention—have transfered remarkably well. I have twice served as a teaching assistant in courses for gifted youth, once for an intensive summer math program for which I was responsible for a great deal of one-on-one interaction and lesson planning. I have also TAed at the college level for a CS theory course, planning and running quiz section, holding office hours, and grading papers, an experience which developed my ability to participate in the planning of a curriculum beyond a one-off lesson.

At present, in addition to occasional math tutoring, I am a leader of a Russian-style math circle for elementary school students. My co-leaders and I design series of problems that avoid the usual elementary-school fare, aiming instead to develop mathematical inquisitiveness and (eventually) maturity. I have also guest-lectured in a high-school computer science classroom, for which I designed a lesson encouraging students to take a step back from their day-to-day programming and think critically about programming-language design.

I propose to expand these outreach activities as a graduate student. I have the teaching and curriculum-design experience to be a leader in a math-circle program, and I will seek mentors in existing outreach programs to develop my skills as a coordinator. It is important, too, that we do not restrict our attention to schools where gifted education is already in place: I propose to identify schools which are underserved by existing gifted programs as targets for math circles, a project which will serve in the long run to mitigate underrepresentation in STEM fields.

It is also important that outreach not stop at mathematics. Computational thinking has become essential to success in STEM fields, and there is no reason for students to wait until college to be introduced to a computer-science curriculum. To this end, I will seek opportunities to guest-teach in middle-and high-school classes and introduce students to computational thinking. This could be as simple as discussing something like stable matching, or, over the course of a few classes, basic programming in Scratch or Processing.

Although I recognize that to be a great teacher like those who set me on my present course is difficult when working within the limitations of an outreach program, I nonetheless aim in the long run to be for others what those few teachers were for me: coaches, role models, and guides. Receiving an NSF Fellowship would enhance my ability to devote time and energy to outreach activities, increasing opportunities for me to help students discover computing and mathematics not as memorization and plug-n-chug but rather as filled with opportunities for exploration and discovery.