Teaching Statement

Salil Vadhan

September 2006

1 Introduction

What I find exciting and rewarding as a professor is not only the pursuit of knowledge through research, but the sharing of that knowledge through teaching and advising. Looking back to my own experiences as an undergraduate here, I recognize the great impact that faculty can have on students' lives. It was the inspiring courses and advising of Profs. Harry Lewis, Michael Rabin, and Les Valiant that drew me into theoretical computer science and set me on the career path that I have followed to this day. Thus I was delighted when I returned to Harvard as faculty and could start participating in its educational mission from the other side.

I believe that courses in the theory of computation can and should be exciting and useful to students pursuing a wide variety of concentrations and careers, by exposing them to an intriguing set of issues, giving a unique computational perspective on the world in which we live, and training them in mathematically rigorous thought. These have been major goals in the courses I have taught and developed, given at least as much weight as the specific subject matter being covered. I also place a strong emphasis on the shared educational mission between the students and me, whereby we are all striving for the class to gain as much as possible from the experience, and all course activities and assignments are clearly directed at this goal. Students seem to have responded well to my approach, as reflected in the CUE evaluations (enclosed in dossier) and the *Phi Beta Kappa Award for Excellence in Teaching* I have received.

In addition to classroom teaching, I am also committed to my role as an advisor for both undergraduate and graduate students and to improving education at the institutional level. My research advising is discussed in more detail in Section 3. At the institutional level, I have been active as a member of the Committee on Undergraduate Education for the past two years and have the main faculty member involved in the developed of a new CUE course evaluation form.

2 Courses

Here I provide brief descriptions of the courses I have taught and developed. Full syllabi are enclosed in the dossier.

CS 120: Introduction to Cryptography. This is a new, undergraduate course in cryptography that I have developed. My aim has been to take the modern, complexity-based approach to cryptography, which is normally taught only at the graduate level, and distill its main ideas in a way that is appealing and accessible to undergraduates. Specifically, I want students to come away knowing how to think precisely about cryptographic security goals, how to reason about such goals, and to understand what various cryptographic primitives do and do not provide. Many security holes in real-life systems are based on misunderstanding these principles, and to remedy this, we must teach these principles not just to future researchers (as is typically done), but to a broader audience including those who go directly into industry after college. I believe that CS 120 has been reasonably successful in achieving this goal. Particularly in the second offering, the course managed to attract a fairly diverse audience, not just the top theory students. The students seemed to enjoy the course, and one student who has gone on to work in the computer security industry has reported that he uses the skills from the class “almost daily.” In terms of the computer science curriculum, it provides adds a focused elective to our theoretical offerings. The other theory courses (CS 121, CS 124, AM 107) all cover ‘core’ material and thus are constrained in how deeply they can delve into a single topic.

CS 121: Introduction to the Theory of Computation. I was thrilled to have the opportunity to teach CS 121 the past two years, when Harry Lewis, who taught it for many years, was occupied with a sabbatical and a freshman seminar. It is a central course in the computer science curriculum, introducing students to the theoretical underpinnings of the field and serving as a prerequisite for many other courses. In addition, the issues it explores are of interest to students in many other fields, from mathematics to linguistics to philosophy, and it can even draw some of these students into computer science by showing them the beautiful theory at its core. (Indeed, it is what attracted me to computer science 14 years ago.) Thus I made an extra effort to attract and address non-concentrators in my two offerings of CS 121, and plan to do even more the next time. I would be very happy to continue teaching CS 121 every year or two, depending on coverage and our curricular needs.

CS 225: Pseudorandomness. This is a new graduate course I have developed to synthesize and convey the unified theory of pseudorandomness that I and others have been developing over the past few years (as discussed in my research statement). Even though this is an advanced graduate course, it has been fairly popular, attracting 32 and 19 students in its two offerings, plus a significant number of auditors. My lecture notes for this course, which I posted on the web, have subsequently been used as the basis of similar courses at several other universities, and I plan to start expanding these notes into a textbook when I offer the course again this Spring.

CS 221: Computational Complexity. This is a core graduate theory course, containing essential material for anyone doing research in theoretical computer science, yet being of potential interest to any mathematical scientist who has seen the prerequisite material in CS 121. I have had the opportunity to teach CS 221 twice, during Les Valiant's sabbaticals. Both times, I have augmented the course with some additional content and have shared the materials with Les.

CS 229r: Topics in the Theory of Computation. This is a new generic course number that I introduced to enable us to offer seminar-style courses to cover current research topics in theoretical computer science that are outside the scope of our current offerings. In Spring ‘05, I used it to cover a collection of intertwined topics involving some of the most exciting recent developments in theoretical computer science (the complexity of approximation problems, particularly involving high-dimensional lattices or cuts in graphs; probabilistically checkable proofs; fourier analysis; low-distortion embeddings of metric spaces; and algebraic computation). In Spring ‘06, my postdoc Dan Gutfreund used it to offer a course on “space-bounded computations” as part of his Applied Math Lectureship (in addition to teaching AM 107).

Future Teaching. I am grateful to the Division for the flexibility and freedom it has given me in teaching and developing the above courses. In addition to continuing to teach these, I also look forward to covering or developing additional courses in the future within the scope of our curricular needs. I would enjoy teaching any of our other theoretical computer science courses (CS 124, CS 22x) and would be happy to get more involved in our applied math offerings (AM 106, AM 107, ...). I also hope to contribute to the efforts in both computer science and applied math to reach out more to preconcentrators and nonconcentrators. As discussed above, I believe that the material in CS 121 can play a role in this effort, but it still seems to be largely perceived as a course for students already in the computer science concentration. It is unclear whether this is only a problem of perceptions or we need to repackage the material to make it more attractive or accessible. In either case, I believe that we have something valuable to offer a broader population of students, and I look forward to working with my computer science and applied math colleagues to make that happen.

3 Research Advising

The kind of theoretical research I do tends not to generate many experiments, programming projects, or calculations that require research assistants. Thus, for me the value of research advising comes from the rewarding experience it provides me as an educator and from the inspiration and fresh perspective on research that comes from collaborating with students. As a consequence, I rarely assign projects to my advisees, whether they be undergraduates, graduate students, or postdocs, and rather jointly work with them to find projects that match their interests and goals. This process naturally means that it takes more time for the student to focus on a specific target, but as a result, almost all of the students who have worked closely with me have been successful in their research (leading to excellent theses and/or papers published in top venues).

A list of all of the students and postdocs I have supervised is contained in my CV, so I will provide only a brief summary here.

Graduate Students. This past Spring, I graduated my first two Ph.D. students. One, Emanuele Viola, had an outstanding thesis on the topic of pseudorandomness, including a paper that won the 2006 Student Paper Award from the Society of Industrial and Applied Mathematics (SIAM). For the coming year, he has secured a membership in the School of Mathematics at the Institute for Advanced Study in Princeton, which is the most coveted postdoctoral position in computational complexity theory (and increasingly for theoretical computer science in general). My other graduated student, Minh Nguyen, wrote an excellent thesis on zero-knowledge proofs, containing the work that led to our new paper in FOCS '06, which solves a long-standing open problem in the area. After much consideration, she has decided to explore other, non-research career possibilities after graduation, but she knows she has my full support should she decide to return. I am currently advising two other Ph.D. students and one S.M. student, and all of their research is going quite well for the stage of the studies.

In addition to my own students, I have tried to foster a stimulating environment for all of the students in the Theory of Computation group, by organizing a weekly seminar through which they can remain abreast of current research developments as well as present their own work, and by inviting numerous visitors and postdocs with whom they can interact.

Undergraduates. I have closely supervised research for four undergraduates (one from MIT) plus a current rising sophomore. Of these four, three have had papers published in top venues, two have won Hoopes prizes, and three have pursued Ph.D.'s in theoretical computer science at top departments. After the two students who graduated in 2004, there was a bit of a gap before I acquired my current advisee, perhaps due partly to me teaching courses that are either further from current research (CS 121) or are too advanced for undergraduates (CS 229r). But I hope that with teaching CS 120, 221, and 225 again, I will enjoy more opportunities to guide our outstanding undergraduates in research.

Postdocs. Through a generous start-up package from Dean Venky, my own research grants, and various institutional sources of funding (CRCS, Applied Math Lectureships, Radcliffe), I have managed to host postdocs for most of my time at Harvard. These postdocs bring energy and activity to the Theory of Computation group, and in particular provide another source of advice and collaboration for our graduate students.

Future Plans. I take a hands-on approach to research advising, in that I meet my advisees on a regular schedule and try to remain actively engaged in the details of their work regardless of whether it is in collaboration with me. Thus, I plan to maintain a research group of modest size, typically consisting of 2–4 graduate students, 1–2 undergraduates, and possibly a postdoc or other visitor. Together with the rest of the Theory of Computation group, this makes for a critical mass of activity that I find stimulating and enjoyable.