Teaching Statement

Elliot Anshelevich

Proposed Teaching

Teaching is one of the main reasons why I would like to work in academia. I greatly enjoy explaining interesting concepts to others, especially energetic undergraduates. I am very interested in teaching at both undergraduate and graduate levels, as there are many concepts, both low-level and high-level, which I am excited about.

At the undergraduate level I am open to teaching a variety of courses, although I would be especially interested in teaching courses on algorithms, data structures, or discrete math. This would be a great opportunity to inspire undergraduate interest in theoretical computer science, and to provide the more applied students with a solid foundation. I believe that it is crucial for all computer science undergraduates to be familiar with these topics, and I would be glad to design new courses of this sort or add content to existing ones. I would also be very happy teaching such undergraduate courses as logic, the basics of computability and complexity, or introductory programming. In addition, I can picture myself teaching slightly more applied courses that contain a large theoretical component, such as undergraduate databases. No matter which course I end up teaching, I will spend time convincing my students of the relevance of the course material to practical applications, as I believe this is necessary to engage the interest of many undergraduates.

Since my own research is in the design and analysis of algorithms, I am well prepared to teach a graduate course on the more central areas of algorithms. In fact, there are two possible courses I could teach on general analysis of algorithms. One is a graduate course covering the important research in the field, that undergraduate courses usually do not get to, such as multicommodity flow, primal-dual methods, and randomization. This course should have regular homeworks and possibly a research project. The other course would be more of a seminar addressing the crucial algorithmic papers from the past decade. If a more narrow algorithmic course is desired, I would be very interested in teaching a course specifically about approximation algorithms and techniques used in the field. Most of my research relies heavily on these techniques, but I also believe that they are some of the most important tools needed by theory graduate students today. If allowed to teach approximation algorithms, I would almost certainly use the book by Vijay Vazirani, and complement it with homework sets I design.

Besides general courses in algorithms, I also look forward to teaching some courses on more focused topics. A large part of my research is in algorithmic game theory, which is becoming more important through the emergence of various systems involving the interaction of strategic agents. I would like to design and teach a course on algorithmic game theory, where I would cover the seminal papers in this area over the past few years, and address such topics as auctions, mechanism design, and networks with strategic agents. Another course I would like to teach at some point in the future is a graduate course on the structure of information networks. This course would also be a survey of important work in the area, and would resemble a course designed by Jon Kleinberg, except I would put more emphasis on influence propagation in networks and the design of gossip protocols. Finally, every few years I plan to teach a seminar in an area I am only somewhat familiar with, as I consider this an excellent way of learning new things. One topic of such a seminar will certainly be the connections between network research in computer science and other fields, such as sociology and physics.

Teaching Experience

My main teaching experience comes from Spring 2004, when I was a teaching assistant in Cornell University for an undergraduate-level class COMS 482, entitled "Introduction to Analysis of Algorithms". This class had around 130 students, weekly homeworks, and covered such things as greedy algorithms, flows, dynamic programming, NP-completeness, and a few more advanced techniques. The class had 2 midterms and a comprehensive final exam.

My responsibilities as a teaching assistant included holding weekly office hours, grading, and proctoring exams. As the senior TA of the course, I also had to generate homework solutions, write up the frequent mistakes that the students made on homeworks, and answer most of the emails sent to the class account. I found the office hours very enjoyable and energizing, as the satisfaction of watching students understand something they did not before, and become interested in the subject, is one of the reasons I decided to pursue a career in academia. I believe I explained the topics well without giving away the answers to homework problems, and I think my students shared this opinion, as the attendance of my office hours kept increasing throughout the semester, with more and more "regulars".

Since at this point I was already a 4th year graduate student, and I knew that I wanted to teach in the near future, I took this opportunity to observe how this large class was run and organized, and ask the instructor as many questions as possible about the organization of the course and his philosophy of teaching. Because of this, I thoroughly understand the work involved in running such a large course with weekly homeworks, and in making sure that everything, including upcoming lectures, grading, re-grading, student requests, and dealing with TA's, proceeds smoothly.

At the end of the semester, I received the "Teaching Assistant Award for Excellence" from the Department of Computer Science for my work. This course was taught by Jon Kleinberg, an excellent teacher who received the Fiona Ip Li 78 and Donald Li 75 Excellence in Teaching Award in 2000.