TEACHING STATEMENT

XIAOKANG QIU

I teach mainly because I always enjoy helping others with those valuable things I have learned. One of the greatest rewards in my life is to be part of a student’s “Ah-ha” moment. I am eager to share the knowledge we have accumulated as computer science researchers and practitioners with students. I am also excited about the prospect of seeing graduate students grow up from students to colleagues. My experiences as a teacher and mentor as well as a student have shaped my philosophy and strategies for teaching.

TEACHING EXPERIENCES

My first academic teaching experience began when I was a Master’s degree student at Nanjing University in China. I worked as a teaching assistant for the undergraduate course “Symbolic Logic” in Fall 2004. It was purely theoretical and essentially a mathematics course. During my PhD study at UIUC, I acquired extensive teaching experience as a teaching assistant for the introductory course “Introduction to Computing for Non-Tech Students” (UIUC CS105, Fall 2009), which was practical, focusing on hands-on learning with important concepts and useful tools.

In the following several semesters, I was fortunate to assist two excellent teachers, Mahesh Viswanathan and Madhusudan Parthasarathy, for “Theory of Computation” (UIUC CS373, Spring 2010, Fall 2010 and Fall 2011). For this kernel course in computer science, I led weekly discussion sections and substituted for the instructor and gave lectures when he was absent. The overall quality of my course was highly rated by students. The score was significantly higher than the average of both UIUC and the CS department. My experiences also gave me opportunities to teach as the head TA for Fall 2010 and Fall 2011. In these two semesters, I played a central role in course development, in designing teaching materials, and in supervising graders and evaluating their performance. Many of those materials we developed are still in use at UIUC.

Now as a postdoc at MIT, I also gain valuable experience in mentoring students on their research. I have been mentoring two undergraduate students in Spring/Fall 2014, respectively, through MIT’s Undergraduate Research Opportunities Program (UROP), and one prodigy student from Colorado School of Mines when he was a research intern at MIT in Summer 2014. They worked on projects such as extending Sketch to support algebraic data types, encoding natural proofs into a synthesis-enabled language, improving program synthesis through the use of packages, etc. Our collaboration has been fruitful: One project has won the first place prize in the Student Research Competition (SRC) at SPLASH 2014; another project has resulted in a paper draft submitted to PLDI 2015. During the course of mentoring these students, I have learned the skills of conveying my research ideas to students and setting concrete goals for them.

In Spring 2015, in response to the lack of a computer-aided programming curriculum at MIT, I collaborated with Prof. Armando Solar-Lezama on designing and instructing MIT’s first Program Synthesis course, “Introduction to Program Synthesis” (MIT 6.885). To our knowledge, the only similar course before ours was UC Berkeley’s “Program Synthesis for Everyone” (UCB CS294). Unlike their syllabus, which focused on constraint-based synthesis, ours provided a comprehensive introduction to the field of program synthesis. This course discussed different techniques for each important theme in program synthesis, including how to define the specifications, how to define the search space, how to search, the interaction models, etc.
Teaching Philosophy

My philosophy of teaching has evolved over the years of teaching and the two decades of being taught and contains some central themes. First, I believe that the most important thing to teach is not the knowledge, but the motivation. I believe that interest is the best teacher, and I firmly believe that computer science is interesting, exciting, and potentially fun. I always enjoy sharing my passion for computer science with my students. I believe that my primary responsibilities as a teacher are threefold: to show the elegance and beauty of computer science, to demonstrate its applicability to solving real-world problems, and to get them excited about it. I also believe that the ability to articulate the motivation of their own work is a necessary precursor for success in their course projects or research projects, and this is true not only for introductory courses but also for advanced courses and graduate work.

Second, I believe that providing intuitions and emphasizing the essentials is crucial. “One picture is worth a thousand words”; this is what I have learned from my PhD advisor, Madhusudan Parthasarathy. He taught several theoretical courses for many years and heavily utilized his computer tablet for visualizing his presentation. Logic and mathematics were significantly involved in these courses, and were fairly difficult to grasp for many students. When he demonstrated concepts and algorithms with intuitive examples and figures, the main ideas immediately became clear to the students.

Finally, I believe that each student has unique needs. Some students learn mostly from hands-on experiences and group discussions, other students prefer reading the text on their own, yet others learn best by one-on-one communication. To the extent possible and practical, I feel that it is important to individualize instruction so that I can meet those particular needs. Office hours have been extremely important for me, as they provide a more interactive venue for asking questions and the opportunity for me to adapt my way of explaining and tailoring the explanation to the student. I believe this kind of individualization will help us engage students, especially in the era of MOOCs.

Future Teaching

As a professor, I would like to import to my new institution many of the ideas that have proved successful in my teaching at UIUC and MIT. Apart from introductory courses, I would be particularly interested in teaching undergraduate and graduate courses in Data Structures, Algorithms, Logic, Compilers, Programming Language Design and Software Engineering. I would also be prepared to develop and teach advanced graduate courses such as Program Analysis and Verification, Program Synthesis, Programming Language Semantics, and seminars related to my research areas.