

# Teaching Statement

**Brian Milch**

Among my reasons for pursuing an academic career, teaching and mentoring students are on a par with doing fundamental research. I had my first teaching experience as an undergraduate at Stanford, where I served for five quarters as a “section leader” for introductory programming classes. This involved teaching weekly tutorial sections and meeting one-on-one with students to provide feedback on their programs. During my senior year at Stanford, I was a teaching assistant for CS 221, the foundational artificial intelligence (AI) course for graduate students and advanced undergraduates. I also TA’ed the undergraduate AI course for two semesters at Berkeley: across these two semesters, my average teaching effectiveness score (given by students in my section) improved from 4.3 to 4.7 out of 5. As a step toward teaching my own courses, I took an elective class in Fall 2005 called “CS 302: Designing Computer Science Education”, in which I designed a course syllabus, problem sets, and exams. I also enjoyed working with Berkeley High School students once a week as a volunteer writing coach from Fall 2004 through Spring 2006.

These experiences, along with readings on computer science education in CS 302, have refined both my teaching skills and my philosophy on teaching. I believe the instructor’s first duty is to convey enthusiasm for the subject, and make it clear how the course topics relate to each other and the real world. Of course, it is important to explain things clearly — students and colleagues have told me that I have a knack for explaining complex material. But I have also learned that explanations are only one aspect of teaching. Another vital skill is listening: students’ questions may not be crisp or well-phrased, but making an effort to understand and answer them can head off confusion. Finally, in many cases, students learn by doing. Well-crafted problem sets and projects are perhaps the most important aspect of a course.

I am very well qualified to teach general courses in artificial intelligence, as well as graduate courses in probabilistic graphical models and statistical machine learning. I could also teach introductory programming, discrete mathematics, algorithms, and logic.

I am also looking forward to advising students, helping them to do important research and to develop as scholars and leaders. As a graduate student at Berkeley, I mentored three undergraduates — David Sontag, Daniel Ong, and Andrey Kolobov — who contributed to our group’s research and became co-authors on our papers. I was struck by the improvement in their skills, confidence, and research judgment over the years that I worked with them. As a post-doc at MIT, I helped advise a master’s student, Ashwin Deshpande, on a project that led to a paper at the Conference on Uncertainty in Artificial Intelligence. I also mentored two undergraduates, Zuoyu Tao and Michael Haimes, through MIT’s Undergraduate Research Opportunities Program. I have begun to learn the art of suggesting research tasks that match each student’s abilities and motivations.

In addition to training the next generation of computer scientists, I am interested in teaching students who are not computer science majors. Many topics in CS have the potential to be both fascinating and useful to students across the university: notions of abstraction and encapsulation, ways of characterizing the complexity of algorithms, universal machines and the existence of undecidable problems, and so on. Most graduates of top universities are exposed to similarly fundamental concepts in physics, chemistry, and biology, but they often have no idea what computer science involves apart from programming. As a faculty member, I would like to share the essentials of our field with a wider audience.