Teaching Statement

Alexandra Silva

Teaching is hard. Teaching is challenging. Teaching is fun. Teaching is rewarding. Teaching makes you a better researcher. These are all answers, seemingly contradictory, that cross my mind when I am asked: *Why do you teach?*

I love the feeling of solving puzzles, all kinds of puzzles, from mathematical problems that arise in my research to the Sunday crossword to the problem of packing my overfilled suitcase in the most efficient way. Solving a puzzle is challenging, and sometimes frustrating, but it forces one to think outside the box, to be persistent, to be curious. And curiosity is of crucial importance in the learning process. Curious people ask the best questions and this leads to great ideas.

If I had to summarize my teaching philosophy in one sentence, I would choose: I want to teach the students how to be curious, how to face challenging problems with methodology and eagerness.

Many computer science curricula are focusing more on courses that teach the latest technology. Although I do think this is important for motivating the students and increasing their employability, the long term vision requires that we give the students the skills to learn new things, to adapt in a fast evolving world. The world of technology changes rapidly and the tools of today will be outdated by the time the students graduate. We should teach students to be persistent, to pursue solutions to problems, to solve puzzles. We should show them how to work in teams in order to tackle complex problems, and how to collaborate fruitfully. Giving them the satisfaction of solving a problem, alone or in a team, will activate their entrepreneurial skills.

I believe a strong computer science curriculum should contain theory courses that provide students with the skills needed to reason abstractly and grasp complex problems and solutions. Courses on formal languages and automata, on basic algorithms and complexity, and on programming paradigms and logic are, in my opinion, important to help shape and developing the reasoning capabilities of future software engineers. I am fond of the view that in the same way that we accept civil engineers have to learn complex math to keep a bridge from falling, we should see software as the bridges of the future. We should therefore teach our students that it is crucial to know how to develop correct software systems.

It is hard to motivate students for theory courses. It is challenging to keep them engaged lecture after lecture. It is fun to come up with different teaching methods to explain basic concepts. And, as a teacher, nothing is as rewarding as seeing students understanding complex, abstract concepts, sometimes after weeks of struggle and frustration. And at the moment they realize they have actually grasped a new idea or a new construction, their eyes sparkle and make it all worthwhile. It is that spark that fires up their drive to do better, to search for not only a solution, but a great solution that goes beyond what was asked. This is why I love teaching.

Students have this wonderful way of surprising you with unexpected questions. And a lot of those questions have activated my curiosity and led me to new research ideas. The contact with young and energetic minds is for me a source of inspiration.

I try to improve my teaching by listening carefully to the students' feedback during the course as well as critically reading the student evaluations at the end of a course. Learning from one's mistakes is an important path to improvement, and in itself a skill that I believe we should pass on.

Organizing a course is like refereeing a football match: there must be margin for creativity for the players, but a clear indication from the beginning about the rules and who makes them. In the beginning of a course I might be perceived by some as being tough and rigid, but the students appreciate the organization and the clarity. They also quickly realise how much I care about teaching them new things, no matter whether they are stronger or weaker. Everyone in my classroom gets a chance to learn, without compromising the level of challenge for the more advanced students. I adapt throughout a course, I try to read the group and to understand how much they are learning by giving them carefully designed homework assignments. My curiosity on their progress has proven effective in building a trust relationship with the students and resulted in improvements in both the course contents and my teaching capabilities.