

## Teaching Statement

Eric Bodden

As I found out during the past term, I really enjoy teaching. PhD students at McGill University are not usually obliged to teach. Nevertheless, I volunteered as a lecturer for the senior-undergraduate level class Compiler Design (COMP 520), with the aim to find out whether I would enjoy the teaching part that usually comes with an academic career. I particularly enjoyed teaching this class because it demonstrates how formal methods like formal languages and language acceptors can actually be put into concrete use. My duties included delivering course lectures, preparing assignments and examinations, supervising a teaching assistant, grading examinations, and assigning final marks. My experience teaching this course reinforced my belief that an academic career would be most appropriate for me.

Something that became apparent to me during this last term is that a lecturer can only always be a stimulus to learning. While it is the role of the lecturer to select suitable learning material, explain the basic concepts of this material and clarify subtle details, the students themselves have to take primary responsibility for their own learning activities. I believe that the lecturer should motivate and support these private learning activities, but he cannot be a substitute for those.

In my class I provided motivation through a large set of applied examples. While in class we would usually cover the basic concepts of language recognition, type checking, code generation and so forth, the students would put these concepts into practice in a term-long group project. I gave the students a milestone to complete every week, where the achievements at each milestone were individually graded. This clearly motivated students to stay on track, and to reinforce the understanding of the lecture material during the course of the term. At the end of the term, all student groups succeeded in delivering a fully working compiler. I believe that learning theoretic concepts by applying them to concrete problems is a powerful concept; moreover this concept is easy to apply in Computer Science because even the most theoretic concepts have usually found useful applications somewhere.

Another interesting fact that I realized was that by explaining material to undergraduate students—an audience that differs substantially from a professional research audience—teaching gave me a useful new perspective on programming languages, which are key to my personal research.

Previously, I have served as a teaching assistant for the class “Programmierung” at RWTH Aachen University, which teaches basic programming skills to junior under-graduate students. Also, I have been a teaching assistant for a course on optimizing compilers at McGill University. During this time I also held multiple guest lectures on the recent advances in the application of program analysis, including some approaches developed by myself. In general I think that lectures can be a useful tool to get students interested in someones personal research—given of course, that the topic fits.

The breadth of my education and research interests allows me to teach undergraduate courses in software engineering, programming languages, parallel programming, formal verification and compilers, among others. At an advanced graduate level, I would lead seminars related to my research in program analysis. These seminars will cover both classic papers and current state-of-the-art work. I feel very comfortable teaching in both, English and German.

In summary, my teaching experiences have convinced me that teaching is a challenging, yet rewarding, activity. Teaching is a vital part of an academic career. I have seen that I can successfully teach, and I expect to continue to develop my teaching skills in the future. I look forward to sharing my enthusiasm for computer science with both undergraduate and graduate students.