

Teaching Statement

Aleksandar Nanevski

Experience

Lecturer for the freshmen level course “Fundamental Data Structures and Algorithms”, co-taught with Peter Lee, Klaus Sutner and Margaret Reid-Miller. Carnegie Mellon University, Spring 2004. This course introduces the fundamental programming concepts, through the presentation of the supporting theory and the practical applications, using the Java programming language. With Margaret Reid-Miller, I was responsible for the section of non-CS majors, with an enrollment of about 80 students.

Teaching Assistant for the senior level course “Formal Languages, Automata and Computation”. Professor: Frank Pfenning. Carnegie Mellon University, Spring 2000.

Teaching Assistant for the senior/graduate level course “Computational Projective Geometry”. Professor: Dana Scott. Carnegie Mellon University, Fall 1998. This course heavily relied on the symbolic computation system *Mathematica* to graphically illustrate the formal mathematical concepts, and encourage experimentation with them.

Teaching Assistant at the Institute of Informatics, Sts. Cyril and Methodius University, Skopje. I taught the following courses: Introduction to Informatics for Students of Mathematics and Physics (Spring 1996, Fall 1996, Spring 1997), Informatics for Students of Agricultural Engineering (Spring 1996, Fall 1996, Spring 1997), Operations Research (Spring 1997), Sets and Logic (Fall 1996).

Philosophy

Much of my high-school and undergraduate education in math and computer science was based on the principle: “memorize the formulas, and you’ll be fine”. Luckily for me, I have good memory, but I don’t believe that this approach leads to effective and pleasant learning experience for the student.

It was only in graduate school, mostly thanks to my interaction with the great teachers and researchers above, that I realized the essential component differentiating good from bad teaching. Good teaching provides the mental images behind the formal development, and relates them to the real world phenomena. It is supposed to capture the imagination of the student, and provide intuition, motivation and perspective. How does good teaching achieve that? By encouraging experimentation, and hands-on experience. This is by no means intended to criticize the formal component of the teaching endeavor (i.e., definitions, theorems and proofs), but these should complement, rather than motivate the presentation.

Not accidentally, this “intuition first – formalization second” approach seems to have been inherent in almost all the important scientific achievements (see “Visual Complex Analysis” by T. Needham for examples in math and physics). I therefore believe that it is also a good strategy for doing one’s research, as it leads to the choice of important and natural research topics, rather than unmotivated and contrived ones. This illustrates my personal research interests in the semantics of programming languages. Language semantics is exactly supposed to provide the intuition and *models* (both informal and formal) behind the syntactic programming constructs.

This is why I believe that teaching is an essential part of research, and look forward to engage in teaching with the goal of refining my ideas, discovering new ones, and presenting them in a compelling, motivated and natural way.

Future Teaching

At the graduate level, I would like to teach advanced programming languages, semantics of computation, and logic with applications to computer science. At the undergraduate level, I feel comfortable teaching introductory computer science courses, data structure and algorithms, principles of programming languages, compilers, theory of computation, complexity, and logic in computer science.