

# Teaching Statement

Frank Fu

My ideas on teaching are drawn from my experience as a student, teaching assistant and part time assistant instructor. As a graduate student, I took classes from computer science, mathematics and philosophy departments. Various instructors from different departments have provided me with a good impression of teaching. As a teaching assistant, I observed and learned from the instructors on how to organize a class. As a part time assistant instructor, I taught an introductory class on functional programming. I want to adopt the following ideas to teach a class: 1. Use a rigous method to convey important concepts. 2. Invite students to think with me during the class. 3. Assign intellectually interesting homeworks and projects.

## 1 Past Experience

### Teaching introductory functional programming in Haskell

I have the opportunity to teach an introductory class on functional programming using the programming language Haskell<sup>1</sup>. The same class in the previous year was taught in another language called LISP. My task was to teach the class once a week for around 10 weeks, developed the Haskell version of class materials and hold weekly lab sessions. There were about 50 second-year undergraduate students in the class, most of them had some experience on imperative programming but had no experience on functional programming.

The goal of this class was to introduce students to the idea of recursion. We progressed from basic concept of recursion and pattern matching to developing several well-known sorting algorithms in Haskell.

For the lab sessions, I was asked to use it as both hand-on session and the homework grading session. Homeworks were consisted of functional programming problems in Haskell ranging from easy to challenging (the challenging problems were bonus optional problems). The students were required to demonstrate their solutions in front of tutors during the lab session and they would receive feedbacks and grades.

For the lectures, other than using slides to help me demonstrate the fundamental concepts, I also put a lot of emphasis on live coding during the class. I asked students to help me constructing programs during the class as well. This helped students to see the thought process of developing functional programs and understanding the interpreter's feedback messages. One of my personal goals of the lecture was to let the students see the elegance of functional programming, I was very glad that in one of the lectures the students told me the program for the mergesort algorithm is much simpler than the one in an imperative language.

The end result of this teaching experience is very encouraging, it was the first class taught in Haskell in that university. Some students get inspired and they later organized weekly meetups to learn more about functional programming in Haskell.

I also learned a lot from this experience. It was my first time to teach a class in a class room, with most of the class materials and homeworks developed by myself. It helps me improve my on-stage presentation and public speaking. I began to be interested in teaching and thought about the kind of classes I would like to teach once I become a fulltime lecturer myself.

### Teaching assistant on programming languages concepts

I was a teaching assistant for the class *programming languages concepts* twice. It was a large class, there were around 80 students each time. The goal of the class is to introduce important concepts of theorem

---

<sup>1</sup><https://www.haskell.org/>

proving and functional programming. Most of the students are second or third year, they have experience on programming in languages like Java but functional programming is new for them. My tasks as a teaching assistant includes: (1) Attending lectures. (2) Holding office hours. (3) Developing grading script.

I paid extra attention to observe how the class was taught and what the student's reactions were. I learned a lot from this class and the instructor: (1) Live coding in front of the students makes the subject more approachable and enhances the interaction with the students. (2) Interesting programming homeworks help the students to learn the essential concepts. (3) Grading is done automatically, the students submit their homework via version controlled system such as SVN, and they get their grades within a day or two. (4) Online class forum such as Piazza<sup>2</sup> helps a lot in reducing the overhead of answering questions from students.

## Working with undergraduate students

I was very lucky to have the opportunity to work with two very well-motivated undergraduates. Both of these experience are quite rewarding and I learned how to work with undergraduates on a well-defined short-term project.

In the context of Trellys project, I worked with Angello Astorga on implementing abstract machine and verified its property using the Iowa Trellys language<sup>3</sup>. Angello was a second year undergraduate student, he had no formal background in formal reasoning and functional programming, this project was his first summer research project. We met about four times a week to discussed questions he had about both programming in general and the specific syntactic rules of the experimental Iowa Trellys language. Angello managed to finish his project and gave a presentation at the Midwest Verification Days in University of Kansas in 2012.

During my postdoctoral research, I worked with undergraduate student Andrew Pond on implementing a heuristic algorithm to generate intermediate lemmas, which can be used in a functional language that allows the type class resolution to be nonterminating. As a third year undergraduate student, Andrew Pond already had a lot of experience on research and functional programming. After setting up a basic data structure framework and explained the basic idea, he was able to contribute to the code base. We also discussed in the office regularly for any questions or ideas he had. We managed to incorporate his work into the prototype and published a paper jointly<sup>4</sup>.

## 2 Proposed Classes

I plan to continue to make live coding as an essential part for the programming class and experiment different methods to involve student participation. I look forward to teach classes that are related to logic, computation and functional programming languages. I will be comfortable to teach introductory functional programming. I would also be interested in teaching the following classes.

- Logic in computer science (undergraduate level).
- Theory of computing (undergraduate level).
- Discrete structures (undergraduate level).
- Programming languages and compiler concepts (undergraduate level).
- Type theory and lambda calculus (advanced/graduate).

---

<sup>2</sup><https://piazza.com/>

<sup>3</sup>Source code at: [https://github.com/sweirich/trellys/blob/master/lib/sepp/Tests/unittests/ck\\_machine.sep](https://github.com/sweirich/trellys/blob/master/lib/sepp/Tests/unittests/ck_machine.sep)

<sup>4</sup>*Proof relevant corecursive resolution*, Peng Fu, Ekaterina Komendantskaya, Tom Schrijvers, Andrew Pond