Preliminary Exam for Programming Languages

Potential Committee Members

Kenneth M. Anderson
Bor-Yuh Evan Chang
Pavol Cerny
Dirk Grunwald
Matthew Hammer
Sriram Sankaranarayanan

Overview

The purpose of the Programming Languages Preliminary Examination is to provide students the opportunity to demonstrate mastery of the fundamentals in programming languages while gaining depth in a specific sub-area of research. The exam is for students who have identified, in consultation with their advisor, a sub-area of programming languages as a potential area for their dissertation work. Overall, the goal of the exam is for students to demonstrate that they are ready for Ph.D.-level research within the chosen field of study.

Format

The exam will consist of three parts: (1) a 45-minute talk summarizing and critiquing a series of papers in the chosen area of research, (2) an oral exam on foundations, and (3) an implementation project.

The student will work with the committee to select an exam date.

Student Preparation

The exam committee will consist of at least three faculty members from those listed above to serve on a student’s preliminary exam committee. The advisor, in consultation with the student, will invite the additional faculty members. The committee will advise each student about the compilation of his or her individual “prelim” bibliographies. Each committee member will advise on papers from his or her area of expertise.

Months in advance of the exam (at least three is suggested), students are expected to begin compiling a bibliography that include breadth in programming languages and depth in the chosen sub-area. This exercise is intended to encourage students to think about the kind of researcher they want to be and what texts will most inform their work.
The implementation project will supplement the reading and demonstrate technical competency in the implementation skills needed to perform research in the chosen area. The project details should also be coordinated with the exam committee months in advance.

**Literature Critique in Area of Research**

Once the papers for the literature critique have been chosen, the student will have four weeks to read these papers and prepare a 45-minute talk about them. It is expected that the student will follow references in the papers, as appropriate, to give a thorough critique.

The committee will evaluate the talk based on (a) the extent to which the prelim-taker was able to understand these papers and extract the key ideas in the papers; (b) the extent to which the prelim-taker was able to identify the strengths and weaknesses of the papers; and (c) the extent to which the prelim-taker was able to communicate that information in a clear but succinct manner.

**Oral Exam on Foundations**

The oral exam will allow the student demonstrate mastery in the fundamentals of programming languages. The exam questions will be drawn for key concepts in sub-areas, such as language design, language semantics, type systems, language implementation, and program analysis.

Possible course preparation includes core graduate courses in compiler design (e.g., CSCI 5525) and programming language semantics (e.g., CSCI 5535) or their equivalents.

Reference texts include the following:

- Benjamin C. Pierce. Types and Programming Languages.
- Steven Muchnick. Advanced Compiler Design and Implementation.
- Andrew Appel. Modern Compiler Implementation.

The committee will evaluate the oral exam based on the extent to which the student is able to explain concepts raised by the exam questions.

**Implementation Project**

The implementation project will allow the student demonstrate mastery in the implementation skills needed to perform research in the chosen subspecialty (e.g., build analyzers).
The committee will evaluate the project based on (a) whether the project demonstrates the ability to prototype advanced programming language tools and (b) the extent to which the student was able to explain the architecture of the system in a clear and concise manner.