

New course for Fall 2005

6.883: PROGRAM ANALYSIS

TR 11-12:30, room 24-407

<http://pag.csail.mit.edu/6.883/>

Lecturer: Michael Ernst, mernst@csail.mit.edu

Software is becoming ever more complex and difficult to understand, at the same time as it is used ever more pervasively. It is hopeless to understand how software systems work (or why they do not work!) without automated assistance. Programmers need tool assistance during design, implementation, testing, debugging, and modification (“maintenance”). This graduate seminar will investigate a variety of program analysis techniques that address these software engineering tasks.

Static analysis topics include abstract interpretation (dataflow), type systems, model checking, decision procedures (SAT, BDDs), theorem-proving. Dynamic analysis topics include testing, fault isolation (debugging), model inference, and visualization. While the course focuses on the design and implementation of programming tools, the material will be useful to anyone who wishes to improve his or her programming or understand the state of the art.

Students are expected to read classic and current technical papers, actively participate in class discussion, perform small exercises that provide experience with a variety of tools, and complete a team research project. Sample projects will be provided, but students are also free to propose their own, particularly ones motivated by their own problems experienced while programming. Examples include proposing and evaluating a fundamental new technique; developing and assessing new algorithms to replace currently-used ones; translating a methodology to a new problem domain; evaluation of proposed techniques (for instance, via a user study); and applying known techniques to new problem domains, such as operating systems, networks, embedded systems, security, biology, aerospace, etc.

12 units. Grad H and EC credit.

Prerequisite: 6.170 and 6.042, or instructor’s permission. 6.035 or 6.821 is helpful but not required.