Program Analysis Techniques for Efficient Software Model Checking

Chandrasekhar Boyapati

Department of Electrical Engineering and Computer Science
University of Michigan — Ann Arbor

Making software reliable is one of the most important technological challenges facing our society today. Software model checking offers a promising approach to address this problem. It is a formal verification technique that exhaustively tests a program on all possible inputs up to a given size and on all possible non-deterministic schedules, but does so cleverly by detecting similarities in the space of test cases and pruning away large numbers of redundant test cases. This talk presents techniques that significantly improve the state of the art in software model checking of programs that use pointers and linked data structures. We believe our techniques can enable the checking of much larger programs and complex program properties than currently possible.

Biography

Chandrasekhar Boyapati is an Assistant Professor in the Electrical Engineering and Computer Science Department at the University of Michigan. He received his BTech degree in Computer Science from the Indian Institute of Technology (IIT), Madras, and his S M and PhD degrees in Computer Science from the Massachusetts Institute of Technology in 2004. Professor Boyapati’s primary research interest is software reliability which spans the spectrum from programming languages through program analysis to software engineering. He is interested in all approaches for helping programmers write and maintain reliable software. He is particularly interested in developing type systems and software model checking techniques to improve software reliability.

Thursday, 20 April 2006, at 3:30pm
223 Atanasoff Hall

— refreshments served —