University at Buffalo
The State University of New York

Department of Computer Science and Engineering

Presents

James S. Royer, Syracuse University

Wholesale Analysis of Algorithms

A Lisp programmer knows the value of everything,
but the cost of nothing. --- Alan Perlis

Perlis' quip is an overstatement---but not by much. Programmers in functional and object-oriented languages have few tools for reasoning about the efficiency of their programs. Almost all tools from traditional analysis of algorithms are targeted toward roughly the first-order fragment of C. What tools there are from formal methods are interesting, but piecemeal and preliminary.

Type systems are among the most successful and widely used of formal methods. We investigate the extent to which one can do ``automatic'' complexity analysis of programs via refinements of standard type systems. As a modest proof of concept we present AR, a restriction of call-by-value PCF for which the underlying model of computation and complexity is a standard abstract machine. AR's type system controls computations in two ways. The first, based on Bellantoni and Cook's safe/normal distinction, controls growth rates within recursions. The second, based on linear logic, controls branching of recursions. The result is a programming formalism that is poly-time sound (all type-level 1 and 2 programs have polynomial run-time bounds), poly-time complete (all polynomial-time type-level 1 and 2 functions are AR-computable), and reasonably expressive (a large natural class of polynomial-time algorithms can be directly expressed in AR).

We shall discuss AR, its type-system, some example AR-code, its time-complexity semantics, and some possible extensions.

This is based on joint work with Norman Danner.

Thursday, November 18, 2010 3:30 - 4:45 PM

University at Buffalo – North Campus – Norton 112

This talk is free and open to the public

For more information, please email **cse-dept@cse.buffalo.edu** or contact (716) 645-3180