Program Analyses using Newton's Method

Thomas Reps

Abstract:
Esparza et al. generalized Newton's method -- a numerical-analysis algorithm for finding roots of real-valued functions -- to a method for finding fixed-points of systems of equations over semirings. Their method provides a new way to solve interprocedural dataflow-analysis problems. As in its real-valued counterpart, each iteration of their method solves a simpler "linearized" problem.

Because essentially all fast iterative numerical methods are forms of Newton's method, this advance is exciting because it may provide the key to creating faster program-analysis algorithms. However, there is an important difference between the dataflow-analysis and numerical-analysis contexts: when Newton's method is used in numerical problems, commutativity of multiplication is relied on to rearrange an expression of the form \( a \times X \times b + c \times X \times d \) into \( (a \times b + c \times d) \times X \). Equations with such expressions correspond to path problems described by regular languages. In contrast, when Newton's method is used for interprocedural dataflow analysis, the "multiplication" operation involves function composition, and hence is non-commutative: \( a \times X \times b + c \times X \times d \) cannot be rearranged into \( (a \times b + c \times d) \times X \). Equations with the former expressions correspond to path problems described by linear context-free languages (LCFLs).

This talk will present a surprising method for solving the LCFL sub-problems produced during successive rounds of Newton's method. The method applies to predicate abstraction, on which most of today's software model checkers rely, as well as to other abstract domains used in program analysis.

Joint work with Emma Turetsky and Prathmesh Prabhu.

Bio:
Thomas W. Reps is the J. Barkley Rosser Professor & Rajiv and Ritu Batra Chair in the Computer Sciences Department of the University of Wisconsin, which he joined in 1985. Reps is the author or co-author of four books and more than one hundred eighty papers describing his research (see http://www.cs.wisc.edu/~reps/). His work has concerned a wide variety of topics, including program slicing, dataflow analysis, pointer analysis, model checking, computer security, code instrumentation, language-based program-development environments, the use of program profiling in software testing, software renovation, incremental algorithms, and attribute grammars.

Monday, February 26, 2018
Gates Hillman Center 8102
3:00PM – 4:00PM