VIJAY GANESH

MIT Computer Science and Artificial Intelligence Lab,
Room 32G-736, The Stata Center, 32 Vassar Street,
Cambridge, MA, USA - 02139

Phone: (650) 387-9950
vganesh@csail.mit.edu
http://people.csail.mit.edu/vganesh

EDUCATION

Sep, 2007 Ph.D. and M.S. in Computer Science
Stanford University.....Stanford, CA
Thesis Title: Decision Procedures for Bit-Vectors, Arrays and Integers
Advisor: David L. Dill

Jun, 2000 M.S. in Electrical Engineering
Stanford University.....Stanford, CA

Nov, 1994 B-Tech in Electronics and Communication
College Of Engineering, Trivandrum,
University of Kerala.....Trivandrum, Kerala, India

INTERESTS

Software reliability, constraint solvers, automated bug finding, program analysis, formal methods, logic, machine learning for solvers and reliability

RESEARCH EXPERIENCE

RESEARCH SCIENTIST

Oct, 2007 – present, *Research Scientist, CSAIL, Massachusetts Institute of Technology*

BuzzFuzz: A Taint-based Directed Whitebox Fuzzer (Oct 2007 - Present): I developed BuzzFuzz, a dynamic taint-based fuzzing tool that automatically constructs error-revealing inputs for large C programs (ICSE '09). BuzzFuzz has successfully exposed several errors in programs with complex input formats (e.g., movie players and document readers). BuzzFuzz requires as input only the source code of the program-under-test, a set of test inputs on which the program behaves as expected, and a list of program points (determined automatically and/or provided by the user) that BuzzFuzz targets for exposing errors. The output of BuzzFuzz is a set of error-revealing inputs for the program-under-test. A key feature of BuzzFuzz is that it can generate error-revealing inputs that penetrate deep into the semantic core of large programs, unlike many other automated bug finders that typically test shallower parts of programs such as input validation code. Also, BuzzFuzz can construct error-revealing inputs where multiple disjoint regions of the input (e.g., multiple fields in a movie file) take specific values together to trigger an error in programs.

Non-clausal Satisfiability Modulo Theories (April, 2008 – Present): I have worked on the problem of efficiently solving constraints that are not in clausal normal form (CNF). This is an important problem since many reliability and AI tools generate constraints that are typically not in CNF. However, many constraint solvers require the translation of such constraints into CNF. Such translations, though polynomial time, destroy the Boolean structure of the input formula. In this work, we leverage the inherent Boolean structure of the constraints for greater efficiency.

Hampi: A Solver for Regular Expression Equations (Nov, 2008 – Present): Tools that can automatically construct security exploits can be very useful. In this context, I worked on Hampi, a solver for equations over regular expressions. Bug finding tools that automatically construct exploits like SQL injection and cross-site scripting attacks in PHP scripts generate such equations. Hampi solves such equations to automatically generate security exploits.

Machine Learning for Tuning Boolean SAT Solvers (Nov, 2008 – Present): I am currently engaged in a project where we are using machine-learning techniques to tune SAT solvers. It is

well known that many Boolean SAT solvers are very sensitive to certain tunable parameters. In an ongoing collaborative work, I used machine learning techniques to train a classifier that maps select input constraint features to locally optimal Boolean SAT solver parameter values. When the classifier receives an heretofore unseen input, it produces solver parameters such that the solver is typically significantly faster with these new parameters, than with default parameters. The classifier itself is simple and has negligible performance overhead.

THESIS RESEARCH

2000 – 2007, *Research Assistant, Computer Systems Laboratory, Stanford University*

STP: A Decision Procedure For Bit-vectors And Arrays (Nov 2005 - 2007): I designed and implemented STP, an efficient and robust decision procedure for the satisfiability problem of the theory of bit-vectors and arrays, that has been optimized for large constraints encountered in software analysis applications (CAV '07). STP has been used as a prover or a counterexample generator in dozens of research tools including automated bug finding, program analysis, model checking and theorem proving programs. New algorithms based on the *abstraction-refinement* paradigm for arrays were developed and implemented in STP. A solving algorithm for linear bit-vector arithmetic was introduced. STP has been shown to handle very large constraints efficiently. The largest constraint handled by STP is a formula which is 412 Mbytes of text, with 2.12 million 32 bit bit-vector variables, array write terms which are tens of thousands of levels deep, and a large number of linear equalities, which STP solves in approximately 2 minutes on a 3.2GHz box. STP was declared the *co-winner of SMTCOMP 2006 competition* in the category of decision procedures for bit-vectors and uninterpreted functions.

Proof-Producing Decision Procedure For Mixed-Linear Arithmetic in CVC (2002 - 2004): Co-designed and implemented a proof producing, online decision procedure for mixed-linear arithmetic as a component in CVC, a widely used combination of decision procedures. The most interesting part of the work was to extend the decision procedure with a natural deduction style of proof system to produce proofs. The CVC framework used these proofs to drive the backtracking mechanism of the built-in SAT solver.

Deciding Presburger Arithmetic Using Model Checking: Co-designed and implemented an algorithm that translated the satisfiability problem for quantifier-free Presburger arithmetic into the emptiness problem for deterministic finite state automata. The emptiness problem was checked using a model-checking tool. I also compared the effectiveness of various methods to solve the satisfiability problem for conjunctions of quantifier-free Presburger arithmetic formulas. This work brought greater clarity to claims by competing methods for solving the satisfiability problem for Presburger arithmetic.

OTHER RESEARCH

Nov 2005 – 2007, *Research Assistant, Stanford University*

EXE: Automatically Generating Inputs of Death (In collaboration with Cristian Cadar, Dawson Engler and others): EXE is a bug finding tool that has been successfully employed in finding bugs in the Linux kernel, the Berkeley Packet Filter and other open source software. My primary contribution to this work was a logic to capture the semantics of C pointers. Essentially, C pointers were treated as bit-vectors in the logic, operations on C pointers were directly converted to operations on bit-vectors, and program memory was represented as an array of bit-vectors. I also provided theorem prover support to EXE in the form of STP. In fact, the design and implementation of STP was largely influenced by the needs of the EXE project.

2000 – 2004, *Research Assistant, Stanford University*

Combination Results For Many Sorted Theories With Overlapping Signatures (2004-2005): Combining decision procedures of theories whose signatures overlap poses a significant theoretical challenge. I lifted the model-theoretic results by Ghilardi for combining overlapping theories in first-order logic to the case of many-sorted logic. Also, I proved a many-sorted version of the Nelson-Oppen combination result, and two new decidability conditions.

WORK EXPERIENCE

July - Sep 1999, *Intern, SRI International, Menlo Park, CA, USA*

Slicing Java: Designed and implemented a slicing tool for Java programs.

1996-1997, *Software Engineer, Texas Instruments (India) Ltd, Bangalore, India*

Chip Simulator: Designed and implemented an instruction-level accurate simulator for Texas Instruments' C54x DSP chips. Also developed a prototype simulator using the idea of compiled simulation.

1994-1995, *Software Engineer, Larsen & Toubro Limited, Mumbai, India*

Designed and implemented assembly level software for industrial controllers.

PUBLICATIONS

PHD THESIS

Vijay Ganesh. Decision Procedures for Bit-vectors, Arrays and Integers. Stanford University, Stanford, CA. September 2007

REFEREED CONFERENCES

- Vijay Ganesh, Tim Leek, and Martin Rinard. Taint-based Directed Whitebox Fuzzing. In *Proceedings of the 31st International Conference on Software Engineering* (ICSE), Vancouver, Canada, July 2009. To Appear

- Vijay Ganesh and David L. Dill. A Decision Procedure for Bit-vectors and Arrays. In *Proceedings of the 19th International Conference on Computer Aided Verification* (CAV), Berlin, Germany, July 2007

- Cristian Cadar, Vijay Ganesh, Peter Pawlowski, David Dill, Dawson Engler. EXE: Automatically Generating Inputs of Death. In *Proceedings of the 13th ACM Conference on Computer and Communications Security* (CCS) Alexandria, Virginia, USA, November 2006

- Sergey Berezin, Vijay Ganesh, and David L. Dill. An Online Proof-Producing Decision Procedure for Mixed-Integer Linear Arithmetic. In *Proceedings of 9th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (TACAS), Warsaw, Poland, April 2003

- Vijay Ganesh, Sergey Berezin, and David L. Dill. Deciding Presburger Arithmetic by Model Checking and Comparisons with Other Methods. In *Proceedings of the 4th International Conference on Formal Methods for Computer Aided Design* (FMCAD), Portland, Oregon, USA, November 2002

- Ashok Halambi, Peter Grun, Vijay Ganesh, Asheesh Khare, Nikil Dutt and Alex Nicolau. EXPRESSION: A Language for Architecture Exploration through Compiler/Simulator Retargetability. In *Proceedings of the International Conference on Design Automation and Test in Europe* (DATE), Munich, Germany, March 1999

REFEREED JOURNALS

- Cristian Cadar, Vijay Ganesh, Peter Pawlowski, David Dill, and Dawson Engler. EXE: Automatically Generating Inputs of Death. *ACM Transactions on Information and System Security* (TISSEC), Volume 12, Issue 2, Article 10, December 2008

REFEREED WORKSHOPS

- Saddek Bensalem, Vijay Ganesh, et al. An Overview of SAL. In *Proceedings of NASA Langley Formal Methods Workshop* (LFM), Williamsburg, Virginia, USA, June 2000

IN PREPARATION

- Adam Kiezun, Vijay Ganesh, and Michael Ernst. A Solver for Equations over Regular Expressions.
- Philippe Suter, Vijay Ganesh, and Viktor Kuncak. Non-clausal Satisfiability Modulo Theories.
- Rishabh Singh, Joseph P. Near, Vijay Ganesh and Martin Rinard. Self-Tuning Boolean SAT Solvers.

OTHER PUBLICATIONS

- Vijay Ganesh, Sergey Berezin, and David L. Dill. A Decision Procedure for Fixed-Width BitVectors. Computer Science Department Technical Report - CSTR 2007-06, Stanford University, 2007
- Vijay Ganesh, Sergey Berezin, Cesare Tinelli, and David L. Dill. Combination Results for Many-Sorted Theories with Overlapping Signatures. Computer Science Department Technical Report - CSTR 2007-04, Stanford University, 2007
- Vijay Ganesh, Hassan Saidi and Natarajan Shankar. Slicing SAL. Computer Science Department Technical Report - CSTR 2007-08, Stanford University, 2007

PROFESSIONAL ACTIVITIES

International Summer School Organizer: Co-organized an international summer school titled "Combination of Decision Procedures Summer School" at SRI International and Stanford University, Aug 9-12, 2004. (Around 30 researchers attended)

Journal Reviewer: International Journal of Software Tools and Technology Transfer (STTT), 2008

Conference Reviewer: Many formal methods and Electronic Design Automation (EDA) conferences

TEACHING

Apr–Jun, 2006	Teaching Assistant	Automata and Complexity Theory (CS 154), Stanford University
July–Aug, 2004	Teaching Assistant	Introduction to Compilers (CS 143), Stanford University

ADVISING

MS (MIT), 2008	Philippe Suter	Thesis Title: Non-clausal Satisfiability Modulo Theories

INVITED TALKS

MIT Lincoln Lab	02/07/2008	Constraint Solvers for Program Analysis and Bug finding
MIT	11/07/2007	Constraint Solvers for Program Analysis and Bug finding
Coverity Inc.	August 2007	STP: A Decision Procedure for Bit-vectors and Arrays

Prof. Martin C. Rinard
Professor in Computer Science
EECS Department, MIT Computer Science and AI Lab
Massachusetts Institute of Technology
32 Vassar Street, Office 32G-744
Cambridge, MA, USA 02139
Phone: 617-258-6922
Email: rinard@lcs.mit.edu

Prof. David L. Dill (Ph.D. Advisor)
Professor in Computer Science
Computer Science Department, Stanford University
353 Serra Mall, Office Gates 348
Stanford, CA, USA 94305-9030
Phone: 650-725-3642
Email: dill@cs.stanford.edu

Prof. Michael Ernst
Associate Professor in Computer Science
CS & E Department, University of Washington, Seattle
(Previously tenured associate professor at MIT)
PO Box 352350
Seattle, WA, USA 98195-2350
Phone: 206-221-0965
Email: mernst@csail.mit.edu

Prof. Dawn Song
Assistant Professor in Computer Science
EECS Department, Computer Science Division
University of California, Berkeley
Soda Hall, Office 675
Berkeley, CA, USA 94720-1776
Phone: 510-642-8282
Email: dawnsong@cs.berkeley.edu

Dr. Natarajan Shankar
Computer Scientist
SRI International
333 Ravenswood Avenue, Office MS EL256
Menlo Park, CA, USA 94025-3493
Phone: 650-859-5272
Email: shankar@csl.sri.com