VIJAY GANESH
Associate Professor, University of Waterloo, Canada

200 University Ave. West, DC 2530
University of Waterloo, ECE and CS
Waterloo, Ontario, Canada N2L 3G1

Phone: +1-519-888-4567 ext. 32866
Email: vganesh@uwaterloo.ca
Web: https://ece.uwaterloo.ca/~vganesh

EDUCATION

Degree	Institution	Graduation Date
PhD, Computer Science	**Stanford University**, CA, USA	Sep 2007
MS, Electrical Engineering	**Stanford University**, CA, USA	Jun 2000
B-Tech, Electronics and Communication	**College of Engineering, Trivandrum**, Kerala, India	Nov 1994

FULL-TIME ACADEMIC APPOINTMENTS

Research Position	Institution	Dates
Co-Director	**Waterloo Artificial Intelligence Institute, Canada**	Mar 2021– present
Associate Professor (with tenure)	ECE and CS (cross-appointed), **University of Waterloo**, Ontario, Canada	Jul 2018 – present
Assistant Professor	ECE and CS (cross-appointed), **University of Waterloo**, Ontario, Canada	Sep 2012 – Jun 2018
Research Scientist	**Massachusetts Institute of Technology**, Cambridge, MA, USA	Oct 2007 – Sep 2012

VISITING/AFFILIATE ACADEMIC APPOINTMENTS

Research Position	Institution	Dates
Visiting Professor	Computer Science, **Indian Institute of Technology**, Bombay, India	Jan 2020 – Dec 2021
Visiting Scientist (organizer: SAT program)	**Simons Institute** for Theory of Computing, **Berkeley**, CA, USA	Jan 2021 – May 2021
Visiting Professor	Computer Science**, University of Toronto**, Ontario, Canada	Sep 2019 – Feb 2020 Jan 2016 – Mar 2016
Visiting Professor	Chebyshev Labs, **St. Petersburg State University**, St. Petersburg, Russia	Apr 2016 – Jun 2016

RESEARCH INTERESTS

Boolean SAT and SMT Solvers, Machine Learning (ML) for Solvers and Solvers for ML
Testing, Analysis, Verification, and Security of Software and AI Systems
Proof Complexity of Solvers, Theorem Provers, and Formal Methods
Application of Solvers in Engineering, Mathematics, and Physics
Mathematical Logic and Foundations of Mathematics

AWARDS, HONORS, AND DISTINCTIONS

The primary focus of my research is the theory and practice of mathematical reasoning algorithms (in particular, SAT/SMT solvers), as well as their applications in software engineering, security, formal methods, AI, mathematics, and physics. In this context, I have received **more than 30 awards, medals, and distinctions** of various kinds for my research, including an **ACM ISSTA Impact Paper Award 2019 (best paper in 10 years)**, an **ACM Test of Time Award at CCS 2016 (best paper in 10 years)**, a citation for **10-year most-influential papers at the DATE conference in 2008 (best paper in 10 years)**, and more than 10 best paper awards or invited papers. A few of these awards and distinctions are listed below. The rest can be found on my website.

1. **ACM ISSTA IMPACT Paper Award 2019 (Best Paper in 10 Years @ ISSTA conference)**
2. Won two silver medals at the SAT competition 2017, Melbourne, Australia
3. **ACM CCS Test of Time Award 2016 (Best Paper in 10 Years @ ACM CCS conference)**
4. Outstanding Paper at the Annual Computer Security Applications Conference (ACSAC) 2016, Los Angeles, USA
5. Early Researcher Award (ERA) 2016, Ontario, Canada
6. Won two gold medals at the SAT competition 2016, Bordeaux, France
7. IBM Faculty Award 2015, Awarded by IBM TJ Watson Research Center, New York, USA
8. MathCheck paper selected as among Best Papers at CADE 2015, Berlin, Germany
9. Z3str2 String Solver paper selected as among Best Papers at CAV 2015, San Francisco, USA
10. Best Paper Award at the Software Product Lines Conference (SPLC) 2015, Nashville, USA
11. Google Faculty Research Award 2013 (Only 15 winners worldwide in software engineering), Awarded by Google Research
12. Heidelberg Laureate Forum Invitee 2013 (Highly selective conference where young researchers are invited to meet with Fields, Turing, and Abel prize winners. My blog post on the event.)
13. Google Faculty Research Award 2011 (Only 12 winners worldwide in software engineering), Awarded by Google Research
14. ACM SIGSOFT Distinguished Paper Award at ISSTA 2009
15. **Ten-year most influential paper at DATE 2008 (Best Paper in 10 Years @ ACM/IEEE DATE conference)**

RESEARCH FUNDING HISTORY

Funding Program	Project Title	PI	Agency	Funds CAD (my %)	Duration
NSERC Discovery	Machine Learning and Solvers: The Next Frontier	Vijay Ganesh	NSERC	$205K (100%)	Apr 2020-Apr 2025
Amazon AWS	Amazon AWS for Automated Reasoning	Vijay Ganesh	Amazon Inc.	$193K (100%)	Jan 2019-Jan 2021
Ripple	Ripple Faculty Fellow: Security of Smart Contracts	Vijay Ganesh	Ripple Inc.	$120K (100%)	Jan 2019-Jan 2023
Amazon	String Solvers for Security	Vijay Ganesh	Amazon Inc.	$187K (100%)	Jan 2018-Jan2020
NSERC Discovery Extension	Next-generation Constraint Solvers for Software Engineering and Security	Vijay Ganesh	NSERC	$25K (100%)	Apr 2018-Mar 2019
NSERC CRD 2017 (Co-PI)	Accelerating multi-objective combinatorial optimizations via GPUs for Cyber-Physical Systems Design	Krzysztof Czarnecki	NSERC	$150K (25%)	Jan 2017 - Dec 2018
Early Researcher Award 2016	Solvers for Security via a Combination of Machine Learning and Deduction	Vijay Ganesh	Ontario	$150K (100%)	Jun 2016 - Jun 2021
IBM Faculty Award 2015	String Solvers for Mobile and Cloud Security	Vijay Ganesh	IBM	$7.5K (100%)	Dec 2015 - Mar 2017
NSERC Create 2015 (Co-PI)	Product-Line Engineering for Cyber-Physical Systems	Krzysztof Czarnecki	NSERC	$1,650K (8%)	Apr 2015 - Mar 2021
NSERC Discovery 2013	Next-generation Constraint Solvers for Software Engineering and Security	Vijay Ganesh	NSERC	$125K (100%)	Apr 2013 - Mar 2018
Google Faculty Award 2013	From Functional Regressions to Security Testing	Vijay Ganesh	Google	$15K (100%)	Oct 2013 – Oct 2016
Waterloo ECE Starter 2012	University of Waterloo ECE Departmental Seed Funding for Junior Faculty	Vijay Ganesh	ECE, Waterloo	$125K (100%)	Sep 2012 - present
Google Faculty Award 2011	Constraint Solvers for Security Testing	Vijay Ganesh	Google	$60K (100%)	Sep 2011 - Sep 2016
NSF Grant 0905244 2009 (Co-PI)	Exposing and Eliminating Errors at Component Boundaries	Martin Rinard	NSF, USA	$682K (33%)	Oct 2009 - Sep 2012
Total award**				**$3.7M CAD (54%)**	

** All funds are in Canadian dollars. My share is approx. $2 million Canadian dollars or 54% of the total funding amount of $3.7 million dollars.

MAJOR RESEARCH ACCOMPLISHMENTS

As stated above, the primary focus of my research is the theory and practice of SAT/SMT solvers, and their applications to problems in software engineering, security, AI, mathematics, and physics. Below I describe some of my most significant practical and theoretical research contributions in this context.

Important Practical Results

1. Machine Learning for Logic: ML-based Solvers and Provers (2014–present)

SAT and SMT solvers are mathematical reasoning algorithms aimed at solving the **satisfiability problem** for Boolean logic and first-order theories respectively. The Boolean satisfiability (or SAT) problem is rightly considered one of the most important problems in computer science and mathematics. Even though the SAT problem has been shown to be NP-complete, and is considered to be intractable in the worst-case, modern conflict-driven clause-learning (CDCL) SAT solvers are capable of solving real-world formulas with tens of millions of variables and clauses in them. As a consequence, they have found wide applicability in diverse areas such as software engineering, security, AI, mathematics, and physics.

One of the most important questions in SAT research is the following: **"why are solvers so efficient, despite the fact that the SAT problem is NP-complete?"**

In my attempt to answer the above question, my collaborators and I made a set of crucial discoveries:

First, it has been known that solvers can be best understood as algorithms that implement proof systems (e.g., resolution). However, until recently, solver implementations used ad-hoc methods to implement these proof systems. By contrast, we showed that solver design can be dramatically improved in terms of efficiency, extensibility, and quality if solver optimization heuristics (aimed at optimally sequencing and selecting proof rules) are implemented using both online and offline machine learning techniques. The underlying reason for this is that solvers produce copious amount of data as they perform their analysis and proof search, and this data can be leveraged using ML techniques to dynamically optimize the solver's performance.

This idea that SAT solvers are combinations of proof rules (logic) and ML optimization heuristics is one of my key contributions and underpins the design of our award-winning SAT solver, MapleSAT. A version of this solver won two gold medals and two silver medals at the annual SAT competition in 2016 and 2017 respectively and has deeply impacted the field of solver design. Other researchers have adopted and adapted ideas from our work in the context of their solver research.

Second, theorists and practitioners have long held the view that SAT solvers work well for real-world instances because of their inherent structure that can be described parametrically and used to prove upper/lower parameterized complexity-theoretic bounds. While many parameters have been proposed, they all fail in some way or the other to appropriately describe this phenomenon of solver efficiency. After extensive empirical work, my collaborators and I have identified a set of parameters, namely, **hierarchical community structure and mergeability**, that can be shown to empirically explain why solvers are so efficient for real-world instances.

While the definitions of these parameters can get technical, the crucial insight is that these parameters not only capture the structure of real-world instances, but also help explain how solvers are so effective at solving large real-world instances. Specifically, when one considers a certain regime of values for these parameters (corresponding to real-world industrial instances), we see that proof size of such formulas scale polynomially with the size of such formulas. Further, we see that the proof search for such formulas tends to be highly localized to certain parts of the formula structure and thus remains effective. By contrast, the proof search for formulas that have been shown to be hard for SAT solvers (e.g., randomly generated expanders) has been shown to be not automatable (unless some widely held complexity-theoretic assumptions are violated).

2. String solvers for software engineering and security (2009–present)

The problem of designing efficient string solving algorithms is recognized as one of the most difficult in by SMT solver experts. These solvers are a particularly important class of tools in the context of security analysis of string-intensive programs, such as Web applications.

Since 2009, I have co-designed a series of string solving algorithms that are efficient for many security applications, culminating in the co-design and implementation of the Z3str3 and its successor Z3str4 string solver. **My decade-long work on the theory and practice of string solvers led to an ACM IMPACT Paper Award at ISSTA 2019.**

3. **STP Solver for scalable symbolic analysis (2005–2012)**

One of my most impactful practical result is the design and implementation of the STP bit-vector and array solver that **played an important role in the development of scalable symbolic execution, widely considered a breakthrough in software engineering and security**. The STP solver's success is primarily due to several abstraction-refinement style algorithms that I came up with for deciding bit-vector and array formulas. These algorithms have stood the test of time and continue to be used and extended to novel settings. STP has been used to find thousands of bugs in many critical pieces of software including Debian Linux and is currently used by many companies to build in-house security analysis or test generation tools. This work led to an **ACM Test of Time Award at the CCS 2016 conference**.

<u>Important Theoretical Results</u>

1. Parameterized proof complexity of SAT and SMT solvers (2016-present)

As stated above, one of the most important problem in solver research is "why are SAT solvers efficient, even though the SAT problem is NP-complete?". This question can be generalized to many other classes of solvers as well, such as ILP or SMT, that are aimed at solving NP-complete, PSPACE-complete, NEXPTIME-complete, or harder problems.

The fact that there exist efficient algorithms that can solve many classes of real-world instances of problems that are theoretically deemed to be hard suggests a gap between theory and practice. Traditional complexity-theoretic analysis of problems tends to focus on the worst-case and is unsuitable for bridging this gap. Fortunately, theorists have developed many other lines of research, such as parameterized complexity and smoothed analysis, that are powerful tools at explaining why certain algorithms work well in practice, and can be applied in the solver setting as well.

By leveraging tools from (parameterized) proof complexity, my collaborators and I have proved a series of results towards explaining the power of SAT and SMT solvers. While this work is still in progress, the initial results that we have obtained have already proven to be very useful. In particular, we showed that SMT solvers are polynomially-equivalent to Res(T) and Res*(T) proof systems, and SAT solvers are polynomially-equivalent to the merge resolution proof system. These theoretical results leverage the merge parameter that we discovered to empirically to explain why solvers are efficient, thus linking theory and practice.

The ultimate goal of this line of research is to establish parameteric upper bounds on the runtime of solvers such that for a suitable regime of parameter values (that correspond to real-world instances) the runtime of CDCL SAT solvers is polynomial in the size of the input and possibly exponential in the parameter (accounting for both proof size and proof search/automatability). Another goal of this line of research is to theoretically explain why certain solver heuristics work well (e.g., 1UIP clause learning scheme, VSIDS branching heuristics, restarts etc.) in terms of parameters that are found to characterize industrial instances. Further, we are also working towards establishing lower bounds on these heuristics, thus explaining under what conditions they fail to perform well.

2. MathCheck SAT+CAS combination for verification of math conjectures (2014-present)

In many areas of mathematics, computers are increasingly being used to prove, formalize, and check theorems. For certain kinds of conjectures (especially ones involving combinatorial structures and arguments), establishing their veracity can be reduced to the problem of checking the satisfiability of a suitably constructed Boolean formula. Given that SAT solvers are efficient general-purpose search methods, such reductions can be very effective. However, SAT solvers do suffer from an important weakness, namely, that they often lack domain-specific knowledge about many areas of mathematics. Such knowledge or advice can be very useful in pruning the search spaces associated with combinatorial math problems. By contrast, computer algebra systems (CAS) are storehouses of mathematical knowledge, but lack the efficient general-purpose search capabilities of solvers.

This state-of-affairs suggests that by combining the general search capability of SAT solvers with domain-specific knowledge-based advice provided by CAS, in a corrective feedback loop, one can effectively solve a variety of combinatorial mathematical problems that cannot be solved by either approach alone (or possibly by any other method). Indeed, my collaborators and I came to this exact realization that if we appropriately combine the search capabilities of SAT solvers with domain-specific advice from a CAS, in a paradigm we call SAT+CAS, one can solve a large number of math problems. We published our first paper on the SAT+CAS paradigm at the CADE 2015 conference. This paper was selected as among the best papers at the conference and we were invited to publish an extended version of the paper in the FMSD journal. Since then, we have published over a dozen papers, solving a variety of mathematical problems using MathCheck, a SAT+CAS tool that we designed and built.

Another direction we are exploring using the SAT+CAS paradigm is that of solving "concrete" complexity-theoretic questions. In complexity, there are a large number of questions of the following form: does there exist a circuit of concrete size k that can solve a given problem. For example, what is the minimum number of multiplication operations one needs to multiply two 3x3 integer matrices. It is known that the minimum number of multiplications needed to solve the "3x3 matrix multiplication problem" must be greater than 19 (that required a very clever proof), but less than or equal to 23. It is not yet known whether the answer is 20, 21, or 22. These kinds of problems are in essence combinatorial search problems that can be solved using the general-purpose search capability of SAT, aided with domain-specific advice from a CAS.

3. Theories over strings (2009–present)

I have done extensive work in the context of decidability and complexity-theoretic questions for theories over strings. In 2020, my collaborators and I established the PSPACE-completeness of a quantifier-free theory of simple regular expressions, arithmetic over string length, and concatenation, among many other complexity and (un)-decidability results for fragments of theories over strings. Also, in 2017, I proved that the quantifier-free theory of string equations, length function, and string-integer conversion is undecidable. In 2012, I proved that the theory of string equations with a single quantifier-alternation is undecidable.

INDUSTRIAL EXPERIENCE

Oct 2017 – Jun 2019, Founding Advisor, **Quantstamp and Blockchain Development Labs (BDL)**, Toronto, Ontario, Canada: Advisor on security analysis methods

May 1999 - Aug 1999, Research Intern, SRI International, Menlo Park, CA, USA: Designed and implemented a slicing tool for the Java programming language

Feb 1996 - Sep 1997, Software Engineer, Texas Instruments (India) Ltd, Bangalore, India: Designed and implemented instruction set simulators for TI's C54x DSP chips

Nov 1994 - Dec 1995, Software Engineer, Larsen & Toubro (India) Ltd, Mumbai, India: Designed and implemented software for industrial programmable logic controllers

PUBLICATION METRICS

See below for a list of top-tier computer science conference venues where my most important papers have been published. All these venues are ranked highly (i.e., A* or A) by the well-regarded conference ranking website CORE.

Topic	Venue ID	Full Name of Conference Venue	CORE	Number of papers
Artificial Intelligence	AAAI	Association for Advancement of Artificial Intelligence	**A***	7
Formal Methods	CAV	International Conference on Computer Aided Verification	**A***	7
Artificial Intelligence	IJCAI	International Joint Conference on Artificial Intelligence	**A***	3
Software Engineering	ICSE	International Conference on Software Engineering	**A***	3
Computer Security	CCS	International Conference on Computer and Communications Security	**A***	2
Artificial Intelligence	ICML	International Conference on Machine Learning	**A***	1
Automated Reasoning	SAT	Theory and Application of Satisfiability Testing	**A**	10
Automated Reasoning	CP	Principles and Practice of Constraint Programming	**A**	4
Software Engineering	ISSTA	International Symposium on Software Testing and Analysis	**A**	2

PUBLICATIONS

The names of the authors listed below are presented in order of their contributions, with the highest contributor first. The names of my students and postdocs are marked with a '*'. You can also find my publications on DBLP, as well as on Google Scholar.

BOOK CHAPTERS

B1. **Vijay Ganesh** and Moshe Vardi. 2020. **On the Unreasonable Effectiveness of SAT Solvers.** In Tim Roughgarden (Ed.), **Beyond the Worst-case Analysis of Algorithms. Cambridge University Press (CUP).** pp. 547-567.

Ph.D. THESIS, STANFORD UNIVERSITY, CA, USA

T1. **Vijay Ganesh**. PhD Thesis Title: Decision Procedures for Bit-vectors, Arrays, and Integers. Stanford University, Stanford, CA, USA. September 2007. Advisor: David L. Dill.

PEER-REVIEWED CONFERENCE PUBLICATIONS

C1. Joseph Scott*, Trishal Sudula, Hammad Rehman, Federico Mora, and **Vijay Ganesh**. BanditFuzz: Fuzzing SMT Solvers with Multi-agent Reinforcement Learning. In the Proceedings of the 24th International Symposium on Formal Methods (**FM 2021**), Virtual Conference (Originally planned for Beijing, China), Nov 20-26, 2021.

C2. Z3str4: A Multi-armed String Solver. Federico Mora, Murphy Berzish*, Mitja Kulczynski, Dirk Nowotka, and **Vijay Ganesh**. In the Proceedings of the 24th International Symposium on Formal Methods (**FM 2021**), Virtual Conference (Originally planned for Beijing, China), Nov 20-26, 2021.

C3. Murphy Berzish*, Joel Day, **Vijay Ganesh**, Mitja Kulczynski, Florin Manea, Federico Mora, and Dirk Nowotka. String Theories involving Regular Membership Predicates: From Practice to Theory and Back. In the Proceedings of the 13th International Conferences on Combinatorics on Words (**WORDS 2021**), Virtual Conference (Originally planned for Rouen, France), Sep 13-17, 2021.

C4. Murphy Berzish*, Mitja Kulczynski, Federico Mora, Florin Manea, Joel Day, Dirk Nowotka, and **Vijay Ganesh.** An SMT Solver for Regular Expressions and Linear Arithmetic over String Length. In the Proceedings of the 33rd International Conference on Computer Aided Verification (**CAV 2021**), Virtual Conference (originally planned for Los Angeles, USA), July 20-23, 2021.

C5. Chunxiao Li, Jonathan Chung, Soham Mukherjee, Marc Vinyals, Noah Fleming, Antonina Kolokolova, Alice Mu, and **Vijay Ganesh**. On The Hierarchical Community Structure of Practical Boolean Formulas. In the Proceedings of the 24th International Conference on Theory and Applications of Satisfiability Testing (**SAT 2021**). Virtual Conference (originally at Barcelona, Spain), Jul 5-9, 2021.

C6. Joseph Scott*, Aina Niemetz, Mathias Preiner, Saeed Nejati*, and **Vijay Ganesh**. MachSMT: A Machine Learning-based Algorithm Selector for SMT Solvers. In the Proceedings of the 27th International Conference on Tools & Algorithms for the Construction & Analysis of Systems (**TACAS 2021**), Virtual Conference (originally planned for Luxembourg), Mar 27 – Apr 1, 2021.

C7. Laura Graves*, Vineel Nagisetty*, and **Vijay Ganesh**. Amnesiac Machine Learning. Accepted at the 35th AAAI Conference on Artificial Intelligence (**AAAI 2021**). Virtual Conference (originally at Vancouver, Canada), Feb 2 – Feb 9, 2021.

C8. Dhananjay Ashok, Joseph Scott*, Maysum Panju, Sebastian Wetzel, and **Vijay Ganesh**. Logic-Guided Genetic Algorithms. Accepted at the 35th AAAI Conference on Artificial Intelligence (**AAAI 2021**). Virtual Conference (originally at Vancouver, Canada), Feb 2 – Feb 9, 2021.

C9. Curtis Bright*, Kevin K. H. Cheung, Brett Stevens, Ilias S. Kotsireas, and **Vijay Ganesh**. A SAT-based Resolution of Lam's Problem. Accepted at the 35th AAAI Conference on Artificial Intelligence (**AAAI 2021**). Virtual Conference (originally at Vancouver, Canada), Feb 2 – Feb 9, 2021.

C10. Saeed Nejati*, Ludovic Le Frioux, and **Vijay Ganesh**. A Machine Learning-based Splitting Heuristic for Divide-and-Conquer Solvers. In the Proceedings of the 26th International Conference on the Principles and Practices of Constraint Programming (**CP 2020**). Virtual Conference (originally at Louvain-la-Neuve, Belgium), Sep 7 - Sep 11, 2020.

C11. Haonan Duan, Saeed Nejati*, George Trimponias, Pascal Poupart, and **Vijay Ganesh**. Online Bayesian Moment Matching based SAT Solver Heuristics. International Conference on Machine Learning (**ICML 2020**). Virtual Conference (originally at Vienna, Austria), Jul 12 – Jul 18, 2020.

C12. Curtis Bright*, Kevin K. H. Cheung, Brett Stevens, Ilias S. Kotsireas, and **Vijay Ganesh**. Unsatisfiability Proofs for Weight 16 Codewords in Lam's Problem. In the Proceedings of the 29th International Joint Conference on Artificial Intelligence (**IJCAI 2020**). Virtual Conference (originally at Yokohama, Japan in July 2020), Jan 7 – Jan 15, 2021.

C13. Joseph Scott*, Federico Mora, and **Vijay Ganesh**. BanditFuzz: A Reinforcement-Learning based Performance Fuzzer for SMT Solvers. 12th Working Conference on Verified Software: Theories, Tools, and Experiments (**VSTTE 2020**). Virtual Conference (originally at LA, USA), July 20-21, 2020.

C14. Vincent Vallade, Ludovic Le Frioux, Souheib Baarir, Julien Sopena, **Vijay Ganesh**, and Fabrice Kordon. Community and LBD-based Clause Sharing Policy for Parallel SAT Solving. In the Proceedings of the 23rd International Conference on Theory and Applications of Satisfiability Testing (**SAT 2020**). Virtual Conference (originally at Alghero, Italy), Jul 3 – Jul 10, 2020.

C15. Chunxiao Li*, Noah Fleming, Marc Vinyals, Toniann Pitassi, and **Vijay Ganesh**. Towards a Complexity-theoretic Understanding of Restarts in SAT Solvers. In the Proceedings of the 23rd International Conference on Theory and Applications of Satisfiability Testing (**SAT 2020**). Virtual Conference (originally at Alghero, Italy), Jul 3 – Jul 10, 2020.

C16. Joseph Scott*, Maysum Panju, and **Vijay Ganesh**. Logic Guided Machine Learning. In the Proceedings of the 34th AAAI Conference on Artificial Intelligence (**AAAI 2020**), New York City, USA, Feb 7 – Feb 12, 2020.

C17. Curtis Bright*, Ilias Kotsireas, and V**ijay Ganesh**. SAT Solvers and Computer Algebra Systems: A Powerful Combination for Mathematics. In the Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering (**CASCON 2019**), Toronto, Canada, Nov 4, 2019.

C18. Saeed Nejati* and **Vijay Ganesh**. CDCL(Crypto) SAT Solvers for Cryptanalysis. In the Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering (**CASCON 2019**), Toronto, Canada, Nov 4, 2019.

C19. William Zhang*, Sebastian Banescu, Steven Stewart, Leonardo Passos, and **Vijay Ganesh**. M-Pro: Combining Static and Symbolic Analysis for Scalable Testing of Smart Contracts. In the Proceedings of the 30th International Symposium on Software Reliability Engineering (**ISSRE 2019**), Berlin, Germany, Oct 28 – 31, 2019.

C20. Reza Babaee*, **Vijay Ganesh**, and Sean Edwards. Accelerated Learning of Predictive Runtime Monitors for Rare Failure. In the Proceedings of the 19th International Conference on Runtime Verification (**RV 2019**), Porto, Portugal, October 8-11, 2019.

C21. Curtis Bright*, Jeurgen Gerhard, Ilias Kotsireas, and **Vijay Ganesh**. Effective Problem-Solving using SAT Solvers. In the Proceedings of the Annual Maple Conference (**Maple 2019**), Waterloo, Ontario, Canada, Oct 15 – Oct 17, 2019.

C22. Hari Govind*, Yakir Vizel, **Vijay Ganesh**, and Arie Gurfinkel. Interpolating Strong Induction. In the Proceedings of the 31st International Conference on Computer Aided Verification (**CAV 2019**), New York City, New York, USA, July 15-18, 2019.

C23. Adam Kiezun, Philip J. Guo, Pieter Hooimeijer, Michael D. Ernst, and **Vijay Ganesh**. Theory and Practice of String Solvers. In the Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis (**ISSTA 2019**), Beijing, China, July 15-19, 2019.
ACM IMPACT Paper Award at ISSTA 2019

C24. Curtis Bright*, Dragomir Djokovic, Ilias Kotsireas, and **Vijay Ganesh**. A SAT+CAS Approach to Finding Good Matrices: Examples and Counterexamples. In the Proceedings of the 33rd AAAI Conference on Artificial Intelligence (**AAAI 2019**), Honolulu, Hawaii, USA, Jan 27 – Feb 1, 2019.

C25. Joel Day, Paul He*, Florin Manea, Dirk Nowotka, and **Vijay Ganesh.** The Satisfiability of Word Equations: Decidable and Undecidable Theories. In the Proceedings of the 12th International Conference on Reachability Problems (**RP 2018**), Marseille, France, September 24 – September 26, 2018.

C26. Saeed Nejati*, Jan Horacek, Catherine Gebotys, and **Vijay Ganesh**. Algebraic Fault Attacks on SHA Hash Functions using Programmatic SAT Solvers. In the Proceedings of the 24th International Conference on the Principles and Practices of Constraint Programming (**CP 2018**), Lille, France, August 26-31, 2018.

C27. Edward Zulkoski*, Ruben Martins, Christoph M. Wintersteiger, Robert Robere, Jia Hui Liang*, Krzysztof Czarnecki and **Vijay Ganesh**. The Effect of Structural Measures and Merges on SAT Solver Performance. In the Proceedings of the 24th International Conference on the Principles and Practices of Constraint Programming (**CP 2018**), Lille, France, August 26-31, 2018.

C28. Edward Zulkoski*, Ruben Martins, Christoph M. Wintersteiger, Jia Hui Liang*, Krzysztof Czarnecki and **Vijay Ganesh**. Learning Sensitive Backdoors with Restarts. In the Proceedings of the 24th International Conference on the Principles and Practices of Constraint Programming (**CP 2018**), Lille, France, August 26-31, 2018.

C29. Curtis Bright*, Albert Heinle, Ilias Kotsireas, and **Vijay Ganesh**. Enumeration of Complex Golay Pairs via Programmatic SAT. In the Proceedings of 43rd International Symposium of Symbolic and Algebraic Computation (**ISSAC 2018**), New York City, New York, USA, July 16-19, 2018.

C30. Jia Hui Liang*, Hari Govind*, Pascal Poupart, Krzysztof Czarnecki, and **Vijay Ganesh**. An Empirical Study of Branching Heuristics through the Lens of Global Learning Rate. In the Proceedings of the 27th International Joint Conference on Artificial Intelligence (**IJCAI 2018**), Stockholm, Sweden, July 13-19, 2018. IJCAI 'Sister Conference Best Paper Track' Invited Paper 2018

C31. Robert Robere, Antonina Kolkolova, and **Vijay Ganesh**. The Proof Complexity of SMT Solvers. In the Proceedings of the 30th International Conference on Computer Aided Verification (**CAV 2018**), Oxford, United Kingdom, July 14-17, 2018.

C32. Dmitry Blotsky*, Federico Mora, Murphy Berzish*, Yunhui Zheng, Ifaz Kabir, and **Vijay Ganesh**. StringFuzz: A Fuzzer for String Solvers. In the Proceedings of the 30th International Conference on Computer Aided Verification (**CAV 2018**), Oxford, United Kingdom, July 14-17, 2018.

C33. Jia Hui Liang*, Chanseok Oh, Minu Mathews, Ciza Thomas, Chunxiao Li, and **Vijay Ganesh**. A Machine Learning based Restart Policy for CDCL SAT Solvers. In the Proceedings of the 21st International Conference on Theory and Applications of Satisfiability Testing (**SAT 2018**), Oxford, United Kingdom, July 9-12, 2018.

C34. Curtis Bright*, Ilias Kotsireas, and **Vijay Ganesh**. A SAT+CAS Method for Enumerating Williamson Matrices of Even Orders. In the Proceedings of the 32nd AAAI Conference on Artificial Intelligence (**AAAI 2018**), New Orleans, Louisiana, USA, February 2-7, 2018.

C35. Jia Hui Liang*, Hari Govind*, Pascal Poupart, Krzysztof Czarnecki, and **Vijay Ganesh**. An Empirical Study of Branching Heuristics through the Lens of Global Learning Rate. In the Proceedings of the 20th International Conference on Theory and Application of Satisfiability Testing (**SAT 2017**), Melbourne, Australia, Aug 28 – Sep 1, 2017. Among Top Three Best Papers @ SAT 2017

C36. Saeed Nejati*, Zack Newsham*, Joe Scott*, Jia Hui Liang*, Catherine Gebotys, Pascal Poupart, and **Vijay Ganesh.** A Propagation Rate based Splitting Heuristic for Divide-and-Conquer Solvers. In the Proceedings of the 20th International Conference on Theory and Application of Satisfiability Testing (**SAT 2017**), Melbourne, Australia, Aug 28 – Sep 1, 2017.

C37. Murphy Berzish*, Yunhui Zheng, and **Vijay Ganesh**. Z3str3: A String Solver with Theory-Aware Branching. In the Proceedings of the 17th Formal Methods in Computer Aided Design Conference (**FMCAD 2017**), Vienna, Austria, Oct 2-6, 2017.

C38. Saeed Nejati*, Jia Hui Liang*, **Vijay Ganesh**, Catherine Gebotys, and Krzysztof Czarnecki. Adaptive Restart and CEGAR-based Solver for Inverting Cryptographic Hash Functions. In the Proceedings of the 9th Working Conference on Verified Software: Theories, Tools, and Experiments (**VSTTE 2017**), Heidelberg, Germany, July 22-23, 2017.

C39. Sanu Subramanian*, Murphy Berzish*, **Vijay Ganesh,** Omer Tripp. A Solver for a Theory of Bit-vectors and Strings. In the Proceedings of 39th International Conference on Software Engineering (**ICSE 2017**), Companion Vol., Buenos Aires, Argentina, May 20-28, 2017. Pages 124-126.

C40. Martin Ochoa, Sebastian Banescu, Cynthia Disenfeld, Gilles Barthe, and **Vijay Ganesh.** Reasoning about Probabilistic Defense Mechanisms against Remote Attacks. In the Proceedings of 2nd IEEE European Symposium on Security and Privacy (**EuroSP 2017**), Paris, France, Apr 26-28, 2017.

C41. Sebastian Banescu, Zack Newsham*, **Vijay Ganesh**, Christian Collberg, and Alexander Pretchner. Code Obfuscation Against Symbolic Execution Attacks. In the Proceedings of the 32nd Annual Computer Security Applications Conference (**ACSAC 2016**), Los Angeles, USA, Dec 5-9, 2016. Pages 189-200.
Outstanding Paper Award @ ACSAC 2016

C42. Murphy Berzish*, Asif Khan, Atulan Zaman, **Vijay Ganesh**, and Derek Rayside. Manifold: An SMT-Based Declarative Language for Electronic and Microfluidic Design Synthesis. Proceedings of the 26th International Conference on Computer Science and Software Engineering Conference (**CASCON 2016**), Toronto, Canada, Oct 31-Nov 2, 2016. Pages 188-193.

C43. Riyad Parvez*, **Vijay Ganesh**, Glenn Wurster, Joe Kirwin, and Paul Ward. Combining Static Analysis and Targeted Symbolic Execution for Scalable Bug-finding in Application Binaries. Proceedings of the 26th International Conference on Computer Science and Software Engineering Conference (**CASCON 2016**), Toronto, Canada, Oct 31-Nov 2, 2016. Pages 116-127.

C44. Curtis Bright*, **Vijay Ganesh**, Albert Heinle, Ilias Kotsireas, Saeed Nejati*, and Krzysztof Czarnecki. MathCheck2: A SAT+CAS Verifier for Combinatorial Conjectures. In the Proceedings of the 18th International Conference on Computer Algebra in Scientific Computing (**CASC 2016**), Bucharest, Romania, Sep 19-23, 2016. Pages 117-133.
Symbolic Computation + Constraint Solving Track Invited Paper 2016

C45. Ed Zulkoski*, **Vijay Ganesh**, and Krzysztof Czarnecki. MathCheck: A Math Assistant based on a Combination of Computer Algebra Systems and SAT Solvers. In the Proceedings of the 25th International Joint Conference on Artificial Intelligence (**IJCAI 2016**), New York, USA, July 9-15, 2016. Pages 4228-4223.
IJCAI 'Sister Conference Best Paper Track' Invited Paper 2016

C46. Steven Stewart, Derek Rayside, **Vijay Ganesh**, and Krzysztof Czarnecki. Accelerating the General Simplex Procedure for Linear Real Arithmetic via GPUs. In the Proceedings of the 8th Working Conference on Verified Software: Theories, Tools, and Experiments (**VSTTE 2016**), Toronto, Canada, July 17-18, 2016. Pages 129-138.

C47. Jia Hui Liang*, **Vijay Ganesh**, Pascal Poupart, and Krzysztof Czarnecki. Learning Rate Based Branching Heuristic for SAT Solvers. In the Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing (**SAT 2016**), July 5, 2016. Pages: 123-140.

C48. Jia Hui Liang*, **Vijay Ganesh**, Pascal Poupart, and Krzysztof Czarnecki. Exponential Recency Weighted Average Branching Heuristic for SAT Solvers. In the Proceedings of the 30th AAAI Conference on Artificial Intelligence (**AAAI 2016**), Tuscon, Arizona, USA, February 12, 2016. Pages 3434-3440.

C49. Jia Hui Liang*, **Vijay Ganesh**, Ed Zulkoski*, Atulan Zaman, and Krzysztof Czarnecki. Understanding VSIDS Branching Heuristic in Conflict-Driven Clause-Learning SAT Solvers. In Proceedings of the 11th International Haifa Verification Conference (**HVC 2015**), Haifa, Israel, Nov 17-19, 2015. Pages 225-241

C50. Zack Newsham*, William Lindsay*, Jian Hui Liang*, **Vijay Ganesh**, Sebastian Fischmeister, and Krzysztof Czarnecki. SATGraf: Visualising the Evolution of SAT Formula Structure in Solvers. In the Proceedings of the 18th International Conference on Satisfiability Testing (**SAT 2015**), Austin, USA, Sep 24-27, 2015. Pages 62-70.

C51. Ed Zulkoski*, **Vijay Ganesh**, and Krzysztof Czarnecki. MathCheck: A Math Assistant via a Combination of Computer Algebra Systems and SAT Solvers. In Proceedings of the 25th International Conference on Automated Deduction (**CADE 2015**), Berlin, Germany, Aug 1-7, 2015. pages 607-622.
Paper selected for Journal of Automated Reasoning Special Issue on Best Papers at CADE'15

C52. Yunhui Zheng, **Vijay Ganesh**, Sanu Subramanian*, Omer Tripp, Julian Dolby and Xiangyu Zhang. Effective Search-space Pruning for Solvers of String Equations, Regular Expressions and Length Constraints. In Proceedings of the 27th International Conference on Computer-aided Verification (**CAV 2015**), San Francisco, July 20-24, 2015. Pages 235-254.
Paper selected for Formal Methods for System Design Special Issue on Best Papers at CAV'15

C53. Jia Hui Liang*, **Vijay Ganesh**, Venkatesh Raman, and Krzysztof Czarnecki. Why SAT-based Analysis of Large Real-world Feature Models is Easy. In Proceedings of 19th International Software Product Line Conference (**SPLC 2015**), Nashville, Tennessee, USA, July 20-24, 2015. Pages 91-100.
Best paper award at SPLC 2015

C54. Zack Newsham*, **Vijay Ganesh**, Gilles Audemard, Sebastian Fischmeister and Laurent Simon. The Impact of Community Structure on SAT Solver Performance. In the Proceedings of the 17th International Conference on Theory and Application of Satisfiability Testing (**SAT 2014**), Vienna, Austria, July 14-17, 2014. Pages 252-268.
Best student paper award at SAT 2014

C55. Yunhui Zheng, Xiangyu Zhang and **Vijay Ganesh**. Z3-str: A Z3-based String Solver for Web Application Analysis. In Proceedings of 9th Joint Meeting on Foundations of Software Engineering (**FSE 2013**), St. Petersburg, Russia, August 18-26, 2013. Pages 114-124. Acceptance Rate 20%

C56. **Vijay Ganesh**, Mia Minnes, Armando Solar-Lezama and Martin C. Rinard. Word Equations with Length Constraints: What's Decidable? In the Proceedings of the 8th Haifa Verification Conference (**HVC 2012**), Haifa, Israel, Nov 2012. Pages 209-226.

C57. Fan Long, **Vijay Ganesh**, Michael Carbin, Stelios Sidroglou and Martin C. Rinard. Automatic Input Rectification. In the Proceedings of 34th International Conference on Software Engineering (**ICSE 2012**), Zurich, Switzerland, Jun 2012. Pages 80-90. Acceptance rate 21%.

C58. **Vijay Ganesh**, Adam Kiezun, Shay Artzi, Philip Guo, Pieter Hooimiejer and Michael Ernst. HAMPI: A String Solver for Testing, Analysis and Vulnerability Detection. In Proceedings of 23rd International Conference on Computer-aided Verification (**CAV 2011**), Snowbird, Utah, USA, July 14 – 20, 2011. Pages 1-19.
Invited Tutorial Paper at CAV 2011

C59. Karthick Jayaraman, **Vijay Ganesh**, Mahesh Tripunitara, Steve Chapin and Martin C. Rinard. Automatic Error Finding for Access-control Policies. In the Proceedings of the 13th ACM Conference on Computer and Communications Security (**CCS 2011**), Chicago, Illinois, USA, November 2011. Pages 163-174. Acceptance Rate 15%.

C60. Adam Kiezun, **Vijay Ganesh**, Philip J. Guo, Pieter Hoomijier, and Michael D. Ernst. HAMPI: A Solver for String Constraints. In Proceedings of the 18th ACM International Symposium of Testing and Analysis (**ISSTA 2009**), Chicago, USA, July 2009. Pages 105-116. Acceptance Rate 29%.
ACM Distinguished (SIGSOFT) Paper Award at ISSTA 2009

C61. **Vijay Ganesh**, Tim Leek, and Martin C. Rinard. Taint-based Directed Whitebox Fuzzing. In the Proceedings of the 31st International Conference on Software Engineering (**ICSE 2009**), Vancouver, Canada, May 2009. Pages 474-484. Acceptance rate 12%.

C62. Karthick Jayaraman, **Vijay Ganesh**, David Harvison, Vijay Ganesh and Adam Kiezun. jFuzz: A Concolic Whitebox Fuzzer for Java. In the Proceedings of NASA Formal Methods Symposium (**NFM 2009**), Mountain View, CA, USA, April 2009. Pages 121-125.

C63. **Vijay Ganesh** and David L. Dill. A Decision Procedure for Bit-vectors and Arrays. In the Proceedings of the 19th International Conference on Computer Aided Verification (**CAV 2007**), Berlin, Germany, July 2007. Pages 519-531. Acceptance Rate 25%

C64. Cristian Cadar, **Vijay Ganesh**, Peter Pawlowski, David L. Dill, and Dawson Engler. EXE: Automatically Generating Inputs of Death. In the Proceedings of the 13th ACM Conference on Computer and Communications Security (**CCS 2006**) Alexandria, Virginia, USA, November 2006. Pages 322-335. Acceptance Rate 15%.
ACM Test of Time Award at CCS 2016

C65. Sergey Berezin, **Vijay Ganes**h, and David Dill. An Online Proof-Producing Decision Procedure for Mixed-Integer Linear Arithmetic. In the Proceedings of the 9th International Conference on Tools & Algorithms for the Construction & Analysis of Systems (**TACAS 2003**), Warsaw, Poland, Apr 2003. Pages 521-536.

C66. **Vijay Ganesh**, Sergey Berezin, and David L. Dill. Deciding Presburger Arithmetic by Model Checking and Comparisons with Other Methods. In the Proceedings of the 4th International Conference on Formal Methods for Computer Aided Design (**FMCAD 2002**), Portland, Oregon, USA, November 2002. Pages 171-186.

C67. Ashok Halambi, Peter Grun, **Vijay Ganesh**, Asheesh Khare, Nikil Dutt, and Alex Nicolau. EXPRESSION: A Language for Architecture Exploration through Compiler/Simulator Retargetability. In the Proceedings of International Conference on Design Automation and Test in Europe (**DATE 1999**), Munich, Germany, March 1999. Pages 485-490.
Ten-year Most Influential Paper at DATE 2008

PEER-REVIEWED JOURNAL PUBLICATIONS

J1. Curtis Bright*, Ilias Kotsireas, Albert Heinle, and **Vijay Ganesh**. Complex Golay Pairs up to Length 28: A Search via Computer Algebra and Programmatic SAT. Journal of Symbolic Computation (**JSC 2021**). Volume 102, January 2021. Pages 153-172. In Special Issue of JSC 2021.

J2. Curtis Bright*, Ilias Kotsireas, and **Vijay Ganesh**. Applying Computer Algebra Systems and SAT Solvers to the Williamson Conjecture. Journal of Symbolic Computation (**JSC 2020**). Volume 100, Dec 2020. Pages 187-209. In Special Issue of JSC 2020

J3. Sebastian Wetzel, Roger Melko, Joseph Scott*, Maysum Panju, and **Vijay Ganesh**. Discovering Symmetry Invariants and Conserved Quantities by Interpreting Siamese Neural Networks. Physical Review Research (**PRR 2020**), Volume 2, Issue 3, Sep 2020. American Physical Society.

J4. Curtis Bright*, Kevin K. H. Cheung, Brett Stevens, Dominique Roy, Ilias Kotsireas, and **Vijay Ganesh**. A Non-existence Certificate for Projective Planes of Order Ten with Weight 15 Codewords. Springer Journal for Applicable Algebra for Engineering, Communications, and Computing (**AAECC 2020**). Volume 31, Issue 3-4, Mar-Apr 2020. Pages 195-213.

J5. Curtis Bright*, Ilias Kotsireas, and **Vijay Ganesh**. New Infinite Families of Perfect Quaternion and Williamson Sequences. IEEE Transactions on Information Theory (**IEEE Trans. Inf. Theory 2019**). Volume 66, Issue 12, Dec 2020. Pages 7739-7751.

J6. Curtis Bright*, Ilias Kotsireas, and **Vijay Ganesh**. The SAT+CAS Paradigm and the Williamson Conjecture (extended abstract). ACM Communications in Computer Algebra (**ACM CCA 2019**), Volume 52, Issue 3, September 2019. Pages 82-84.

J7. Curtis Bright*, Dragomir Djokovic, Ilias Kotsireas, and **Vijay Ganesh**. The SAT+CAS method for Combinatorial Search with Applications to Best Matrices. Annals of Mathematics and Artificial Intelligence (**AMAI 2019**). Volume 87, Dec 2019. Pages 321-342.

J8. Jianmei Guo, Jia Hui Liang*, Kai Shi, Dingyu Yang, Jingsong Zhang, **Vijay Ganesh**, Krzysztof Czarnecki, and Huiqun Yu. SMTIBEA: A Hybrid Multi-Objective Optimization Algorithm for Configuring Large Constrained Software Product Lines. International Journal of Software and System Modeling (**SOSM 2019**). Volume 18, Issue 2, August 2019. Pages 1447-1466.

J9. Yunhui Zheng, **Vijay Ganesh**, Sanu Subramanian*, Omer Tripp, Julian Dolby, and Xiangyu Zhang. Z3str2: An Efficient Solver for Strings, Regular Expressions, and Length Function. Formal Methods in System Design Journal (**FMSD 2017**), Volume 50, Issue number 2-3, June 2017. Pages 249-288.
Invited Paper in the Formal Methods in System Design Journal 2017

J10. Ed Zulkoski*, Curtis Bright*, Albert Heinle, Ilias Kotsireas, Krzysztof Czarnecki, and **Vijay Ganesh**. Combining SAT Solvers with Computer Algebra Systems to Verify Combinatorial Conjectures. In the Journal of Automated Reasoning (**JAR 2017**), Volume 58, Issue 3, March 2017. Pages 313-339. Published by Springer Nature.
Invited Paper in Special Issue of Journal of Automated Reasoning 2017

J11. Karthick Jayaraman, Mahesh Tripunitara, **Vijay Ganesh**, Steve Chapin, and Martin C. Rinard. Mohawk: Abstraction-refinement and Bound Estimation for Verifying Access-control Policies. In ACM Transactions on Information and System Security (**TISSEC 2013**), Volume 15, Issue number 4, April 2013. Pages 18:1 – 18:28.

J12. **Vijay Ganesh**, Adam Kiezun, Shay Artzi, Phillip Guo, Pieter Hooimiejer, and Michael D. Ernst. HAMPI: A Solver for Word Equations over Strings, Regular Expressions and Context-free Grammars. In ACM Transactions on Software Engineering and Methodology (**TOSEM 2012**), Volume 21, Issue number 4, Nov 2012. Pages 25:1 – 25:28.
Invited Paper in TOSEM Journal 2012

J13. Cristian Cadar, **Vijay Ganesh**, Peter Pawlowski, David L. Dill, and Dawson Engler. EXE: Automatically Generating Inputs of Death. ACM Transactions on Information and System Security (**TISSEC 2008**), Volume 12, Issue number 2, December 2008. Pages 10:1-10:38.

PEER-REVIEWED WORKSHOP PUBLICATIONS

W1. Vineel Nagisetty, Laura Graves, and **Vijay Ganesh**. xAI-GAN: Enhancing Generative Adversarial Networks via Explainable AI Systems. Accepted at **AAAI-xAI 2021**. Virtual Conference (originally at Vancouver, Canada), Feb 2 – Feb 9, 2021.

W2. Joseph Scott, Aina Niemitz, Mathias Preiner, and **Vijay Ganesh**. MachSMT: A Machine Learning-based Algorithm Selector for SMT Solvers. Workshop on Satisfiability Modulo Theories (**SMT 2020**). Virtual Workshop (originally planned for Paris, France), July 5-6, 2020.

W3. BanditFuzz: A Reinforcement Learning based Performance Fuzzer for SMT Solver. Workshop on Satisfiability Modulo Theories (**SMT 2020**). Virtual Workshop (originally planned for Paris, France), July 5-6, 2020.

W4. Curtis Bright, Kevin K. H. Cheung, Brett Stevens, Ilias S. Kotsireas, and **Vijay Ganesh**. Non-existence Certificates for Ovals in a Projective Plane of Order Ten. In the Proceedings of the 31st International Workshop on Combinatorial Algorithms (**IWOCA 2020**). Virtual Conference (originally planned for Bordeaux, France), Jun 8 – Jun 10, 2020.

W5. Gereon Kramer, Erika Abraham, and **Vijay Ganesh**. Proof Complexity of MCSAT Solvers. In the Proceedings of the 4th International Workshop on Satisfiability Checking and Symbolic Computation (**SC^2 2019**), Bern, Switzerland, 10th July 2019.

W6. Ed Zulkoski*, Robert Robere, Jia Hui Liang*, Ruben Martins, Christoph Wintersteiger, Krzysztof Czarnecki, and V**ijay Ganesh**. Relating Complexity-theoretic Parameters with SAT Solver Performance. In Pragmatics of Constraint Reasoning Workshop (**PoCR 2017**), Melbourne, Australia, Aug 28-Sep 1, 2017.

W7. **Vijay Ganesh**, Sebastian Benascu, and Martin Ochoa. The Meaning of Attack-resistant Systems. In Proceedings of International Workshop on Programming Languages and Analysis for Security (**PLAS @ ECOOP 2015**), Prague, Czech Republic, July 6, 2015. Pages 49-55.

W8. Nikolaj Bjorner, **Vijay Ganesh**, Raphael Michel, and Margus Veanes. An SMT-LIB Format for Sequences and Regular Expressions. In the Proceedings of the International Workshop on Satisfiability Modulo Theories (**SMT 2012**), Manchester, UK, June 30, 2012. Pages 77-87.

W9. Raphael Michel, **Vijay Ganesh**, Arnaud Hubaux, and Patrick Heymans. An SMT-based Approach to Automated Configuration. In the Proceedings of the International Workshop on Satisfiability Modulo Theories (**SMT 2012**), Manchester, UK, June 30, 2012. Pages 109-119.

W10. **Vijay Ganesh**, Michael Carbin and Martin C. Rinard. Cryptographic Path Hardening: Hiding Vulnerabilities in Software using Cryptography. Off-the-beaten-track. Co-located with Principles of Programming Languages Conference, Philadelphia, USA, Jan 2012.

W11. Saddek Bensalem, **Vijay Ganesh**, et al. An Overview of SAL. In Proceedings of NASA Langley Formal Methods Workshop (**LFM 2000**), Williamsburg, Virginia, USA, June 2000.

EDITED VOLUMES

E1. Zhe Hou and **Vijay Ganesh**. Automated Technology for Verification and Analysis, 19th **ATVA 2021**, Gold Coast Australia, Oct 18-22, 2021. Lecture Notes in Computer Science 12971, Springer 2021.

E2. Cristian Cadar, **Vijay Ganesh**, Raimondas Sasnauskas, and Koushik Sen. Symbolic Execution and Constraint Solving. Dagstuhl Seminar 14442, Volume 4, Issue 10, **Dagstuhl Reports 2015**.

E3. Armin Biere, **Vijay Ganesh**, Martin Grohe, Jakob Nordström, and Ryan Williams. The Theory and Practice of SAT Solving. Dagstuhl Seminar 15171, Volume 5, Issue 4, **Dagstuhl Reports 2015**.

E4. **Vijay Ganesh** and Nicky Williams. Constraints in Software Testing, Verification, and Analysis (**CSTVA 2014**), ACM 2014. ISBN 978-1-4503-2847-0.

INVITED TALKS

Institution/Venue (significant venues highlighted in boldface)	**Date**	**Talk Title**
1. Goethe, HPI Potsdam, Hamburg, Frankfurt Institute for Advanced Studies (ADYN group)	Dec 2021	On the Unreasonable Effectiveness of SAT Solvers: Logic + Machine Learning
2. McGill University, Montreal, Canada	Nov 2021	On the Unreasonable Effectiveness of SAT Solvers: Logic + Machine Learning
3. Harvard University, Department of Applied Mathematics, Massachusetts, USA	Nov 2021	When Computer Algebra meets Satisfiability: A New Approach to Combinatorial Mathematics
4. Keynote @ FroCos 2021, Birmingham, UK	**Sep 2021**	**On the Unreasonable Effectiveness of SAT Solvers: Logic + Machine Learning**
5. Rice University, Texas, USA	Aug 2021	When Computer Algebra meets Satisfiability: A New Approach to Combinatorial Mathematics
6. SC^2 Workshop, College Station, Texas, USA	Aug 2021	Machine learning and Logic Solvers: The Next Frontier
7. Bowie State University, Maryland, USA	Apr 2021	On the Unreasonable Effectiveness of SAT Solvers: Logic + Machine Learning
8. University of Gottingen, Germany	Mar 2021	On the Unreasonable Effectiveness of SAT Solvers: Logic + Machine Learning
9. Waterloo Crypto Seminar, University of Waterloo, Ontario, Canada	Mar 2021	On the Unreasonable Effectiveness of SAT Solvers: Logic + Machine Learning
10. Waterloo ECE Society, University of Waterloo, Ontario, Canada	Mar 2021	At the Confluence of Logic and Machine Learning
11. Simons Semester on SAT, Berkeley, CA, USA	Feb 2021	Perspectives on Practice and Theory of SAT Solving

12. Waterloo Computational Mathematics Colloquium, University of Waterloo	Oct 2020	Machine Learning and Logic Solvers: The Next Frontier
13. International Congress on Mathematical Software, Braunschweig, Germany	Jul 2020	Machine Learning and Logic Solvers: The Next Frontier
14. Vector Institute, Toronto, Canada	Feb 2020	Machine learning and Logic Solvers: The Next Frontier
15. Indian Institute of Technology, Bombay, India	Dec 2019	Machine Learning and Logic Solvers: The Next Frontier
16. Sorbonne University (previously, UPMC), Paris, France	Sep 2019	Machine Learning for SAT Solvers
17. Hasso Plattner Institute, Potsdam, Germany	Aug 2019	Machine Learning for SAT Solvers
18. Keynote @ International Symposium on Software Testing and Analysis (ISSTA), Beijing, China	**July 2019**	**ACM IMPACT Paper Award: Theory and Practice of String Solvers**
19. Indian SAT/SMT Winter School, IIIT, Hyderabad, India	Dec 2018	SAT and SMT Solvers: A Foundational Perspective (3 lectures)
20. Waterloo AI Institute, Canada	Oct 2018	Machine Learning for SAT Solvers
21. Marktoberdorf Summer School on Dependable and Secure Software Systems, Marktoberdorf, Germany	Sep 2018	SAT and SMT Solvers: A Foundational Perspective (4 lectures)
22. Theory and Practice of Satisfiability Solving, BIRS, Oaxaca, Mexico	Aug 2018	Machine Learning for SAT Solvers
23. Plenary Talk @ The 24th Conference on Applications of Computer Algebra (ACA 2018), Santiago De Compostela, Spain	**Jun 2018**	**Plenary Talk: SAT Solvers and Computer Algebra Systems: A Powerful Combination for Mathematics**
24. University of Kiel, Germany	Apr 2018	Mergeability and the Unreasonable Effectiveness of SAT Solvers
25. East China University of Science & Technology, Shanghai, China	Apr 2018	On the Unreasonable Effectiveness of SAT Solvers
26. Academia Sinica, Taipei, Taiwan	Apr 2018	On the Unreasonable Effectiveness of SAT Solvers
27. National University of Singapore	Apr 2018	On the Unreasonable Effectiveness of SAT Solvers
28. Microsoft Research, Bangalore, India	Jan 2018	On the Unreasonable Effectiveness of SAT Solvers
29. Indian Institute of Science, Bangalore, India	Dec 2017	On the Unreasonable Effectiveness of SAT Solvers

30. CSTVA 2017 Workshop, Melbourne, Australia	Aug 2017	On the Unreasonable Effectiveness of SAT Solvers
31. David Dill @ 60 Workshop, Heidelberg, Germany	Jul 2017	On the Unreasonable Effectiveness of SAT Solvers
32. University of Luxembourg	Jul 2017	On the Unreasonable Effectiveness of SAT Solvers
33. Hasso Plattner Institute, Potsdam, Germany	Jul 2017	On the Unreasonable Effectiveness of SAT Solvers
34. Rice University, Houston, USA	Jun 2017	On the Unreasonable Effectiveness of SAT Solvers
35. University of Wisconsin, Madison, USA	Jun 2017	On the Unreasonable Effectiveness of SAT Solvers
36. University of Washington, Seattle, USA	Jun 2017	On the Unreasonable Effectiveness of SAT Solvers
37. Microsoft Research, Redmond, USA	Jun 2017	On the Unreasonable Effectiveness of SAT Solvers
38. Achim Kempf Research Group, Waterloo, Canada	Jun 2017	On the Unreasonable Effectiveness of SAT Solvers
39. IQC, Waterloo, Canada	Jun 2017	On the Unreasonable Effectiveness of SAT Solvers
40. Inria/Loria, Nancy, France	Apr 2017	On the Unreasonable Effectiveness of SAT Solvers
41. Max Planck Institute (MPI), Saarbrucken, Germany	Apr 2017	On the Unreasonable Effectiveness of SAT Solvers
42. Moscow State University, Moscow, Russia	Apr 2017	On the Unreasonable Effectiveness of SAT Solvers
43. College of Engineering, Trivandrum, India	Sep 2016	Towards a Deeper Empirical Understanding of CDCL SAT Solvers
44. Institute for Mathematical Sciences, NUS, Singapore	Sep 2016	Towards a Deeper Understanding of Branching Heuristics in SAT Solvers
45. Fields Institute, Toronto, Canada	Aug 2016	Towards a Deeper Understanding of Branching Heuristics in SAT Solvers
46. Keynote @ CSTVA, Saarbrucken, Germany	**Jul 2016**	**Towards a Deeper Empirical Understanding of CDCL SAT Solvers**
47. Moscow State University, Moscow, Russia	Apr 2016	Towards a Deeper Empirical Understanding of CDCL SAT Solvers
48. St. Petersburg State, Chebyshev Labs, Russia	Apr 2016	Towards a Deeper Empirical Understanding of CDCL SAT Solvers
49. SRI International, Menlo Park, CA, USA	Jul 2015	Towards a Deeper Empirical Understanding of CDCL SAT Solvers

50. University of California, Berkeley, CA, USA	Jul 2015	Towards a Deeper Empirical Understanding of CDCL SAT Solvers
51. Moscow State University, Russia	Jun 2015	Towards a Deeper Empirical Understanding of CDCL SAT Solvers
52. Dagstuhl, Saarland, Germany	Apr 2015	Towards a Deeper Empirical Understanding of CDCL SAT Solvers
53. Computer Algebra Group, U. of Waterloo, Canada	Mar 2015	An Introduction to SAT/SMT Solvers
54. Dagstuhl, Saarland, Germany	Feb 2015	Impact of Community Structure on SAT Solver Performance
55. Institute of Mathematical Sciences, Chennai, India	Feb 2015	Impact of Community Structure on SAT Solver Performance
56. MapleSoft Inc., Waterloo, Canada	Dec 2014	Solvers for Software Reliability and Security
57. IBM PL Day, TJ Watson Center, NY, USA	Nov 2014	Impact of Community Structure on SAT Solver Performance
58. Dagstuhl, Saarland, Germany	Oct 2014	SAT/SMT Solvers and Applications: Key Ideas and Future Directions
59. Dagstuhl, Saarland, Germany	Aug 2014	Impact of Community Structure on SAT Solver Performance
60. Banff Mathematical Research Station, Canada	Jan 2014	VSIDS Branching Heuristic and Timed Graph Centrality
61. Institute of Mathematical Sciences, Chennai, India	Dec 2013	Undecidability Results for Word Equations, Regular Expressions, and Length Function
62. Aspiring Minds, New Delhi, India	Dec 2013	Solvers for Software Reliability and Security
63. ICCC 2013, College of Engineering, Trivandrum, India	Dec 2013	Solvers for Software Reliability and Security
64. ADDCT 2013, Lake Placid, NY, USA	Jun 2013	Undecidability Results for Word Equations, Regular Expressions, and Length Function
65. Google Inc., New York City, New York, USA	Jun 2013	Solvers for Software Reliability and Security
66. HCSS, Annapolis, Maryland, USA	May 2013	String Solvers for Reliability and Security
67. Kesterel Institute, Palo Alto, California, USA	May 2013	Solvers for Software Reliability and Security
68. Purdue University, IN, USA	May 2013	Solvers for Software Reliability and Security

69. University of Iowa, Iowa City,USA	May 2013	Solvers for Software Reliability and Security
70. Dagstuhl, Saarland, Germany	Feb 2013	SAT/SMT Solvers for Reliability, Security, and Repair
71. Keynote at CSTVA'12, Montreal, Canada	**Apr 2012**	**Solvers for Software Reliability and Security**
72. Dagstuhl, Saarland, Germany	Feb 2012	SAT/SMT Solvers for Reliability and Security
73. Invited Tutorial @ CAV 2011, Snowbird, Utah, USA	**Jul 2011**	**HAMPI: A String Solver for Testing, Analysis & Vulnerability Detection**
74. Dagstuhl, Saarland, Germany	Aug 2011	SAT/SMT Solvers for Software Reliability and Security
75. Dagstuhl, Saarland, Germany	Jul 2011	SAT/SMT Solvers for Software Reliability and Security
76. Dagstuhl, Saarland, Germany	Apr 2010	SAT/SMT Solvers for Software Reliability and Security
77. Massachusetts Institute of Technology Lincoln Labs, USA	Feb 2008	Constraint Solvers for Program Analysis and Bug-finding
78. Massachusetts Institute of Technology, Cambridge, USA	Nov 2007	Constraint Solvers for Program Analysis and Bug-finding
79. Coverity Inc. San Francisco, CA, USA	Sep 2007	The STP Bit-vector and Array Solver

TEACHING

COURSES DEVELOPED

Computer-aided Reasoning for Software Engineering (ECE 750-T28): At the University of Waterloo, I developed a graduate course on computer-aided reasoning (SAT/SMT solvers and proof assistants) and their applications in software engineering, security, AI, and mathematics. The course is divided into three modules: In the first module, I cover the basics of mathematical logic, such as proof systems and their properties (e.g., soundness, completeness), first-order theories, incompleteness, as well as computability and complexity theory. In the second module, I cover modern conflict-driven clause-learning SAT solvers, SMT solvers, their design and architecture, combination of theories, DPLL(T) etc. Finally, in the third module I cover application of SAT/SMT solvers to verification and analysis of software systems, with a special focus on symbolic execution-based testing and analysis. The course also has a project component, and the students are required to build a system based on an existing SAT or SMT solvers (or develop their own solver) as part of their project. The course has been very well received by students and many course projects have been converted into published papers at top-tier conferences.

Discrete Mathematics and Logic (ECE 208): At the University of Waterloo, I developed a theory undergraduate course (ECE 208) for ECE students who may not have had any background in logic, computability, or complexity theory. The course is quite comprehensive in that it covers Boolean logic, first-order logic, computability, and basic complexity theory. There is also a strong focus on the Boolean satisfiability problem and SAT solver algorithms. This course has been very well received by my students and is one of my favorite undergraduate courses.

COURSES TAUGHT

Term and Year	Course Number/Name	Enrollment
Fall 2021 (Graduate)	ECE650 (Methods and Tools for Software Engineering)	194
Fall 2021 (Graduate)	ECE 750-T28 (Computer-aided Reasoning)	12
Spring 2021 (Graduate)	ECE653 (Software Testing, Quality Assurance, and Maintenance)	64
Spring 2021 (Undergraduate)	ECE208 (Discrete Mathematics and Logic)	132
Fall 2020 (Undergraduate)	ECE208 (Discrete Mathematics and Logic)	191
Spring 2020 (Graduate)	ECE653 (Software Testing, Quality Assurance, and Maintenance)	92
Spring 2019 (Undergraduate)	ECE208 (Discrete Mathematics and Logic)	140
Fall 2018 (Graduate)	ECE 750-T28 (Computer-aided Reasoning)	4
Spring 2018 (Undergraduate)	ECE 351 (Introduction to Compilers)	160
Fall 2017 (Graduate)	ECE 750-T28 (Computer-aided Reasoning)	5
Spring 2017 (Undergraduate)	ECE 458 (Introduction to Computer Security)	134
Spring 2017 (Undergraduate)	ECE 351 (Introduction to Compilers)	132
Fall 2016 (Graduate)	ECE 750-T28 (Computer-aided Reasoning)	12
Fall 2015 (Undergraduate)	ECE 250 (Algorithms and Data Structures)	129
Winter 2015 (Graduate)	ECE 750-T28 (Computer-aided Reasoning	7
Winter 2015 (Undergraduate)	ECE 351 (Introduction to Compilers)	69
Spring 2014 (Undergraduate)	ECE 458 (Introduction to Computer Security)	76
Spring 2014 (Undergraduate)	ECE 351 (Introduction to Compilers)	58
Winter 2014 (Undergraduate)	ECE 351 (Introduction to Compilers)	84
Fall 2013 (Graduate)	ECE 750-T28 (Computer-aided Reasoning)	15
Winter 2013 (Undergraduate)	ECE 458 (Introduction to Computer Security)	74

SUPERVISION

From 2011-present, I have (co)-advised and graduated 2 postdoctoral fellows (one was previously my PhD student) and 8 PhD students. Of these, 2 are currently tenure-track or tenured professors, 4 others are researchers at companies such as IBM, Amazon, Microsoft and Quantstamp, and 2 are working at high-tech companies. I have also co-advised/mentored 14 master's and around two dozen bachelor's students.

CURRENT STUDENTS AND POSTDOCTORAL FELLOWS (PDFs)

Degree	Name	Advisor	Start	Research topic	End
PhD	Behkish Nassirzadeh	Vijay Ganesh	05/2019	SAT-based Cryptanalysis	2025
PhD	Joe Scott	Vijay Ganesh	09/2017	Deep learning and SAT solving	2021
PhD	Ian Li	Vijay Ganesh	09/2019	Restarts in SAT solvers	2023
MASc	Alaina Mahalanobis	Vijay Ganesh	09/2020	xAI-based GANs	2022
MASc	Abinaya Venkatesan	Vijay Ganesh	09/2020	SAT Solvers for Exact Circuit Complexity	2022
MASc	Tony Pan	Vijay Ganesh	09/2021	Fuzzers for DNN Solvers	2022
MASc	John Lu	Vijay Ganesh	01/2022	String Solvers	2022

GRADUATED STUDENTS AND POSTDOCTORAL FELLOWS (PDFs)

Degree	Name	Lead Supervisor	Co-supervisor	Grad date	Research Topic	Current Status
PDF, Waterloo	Curtis Bright	Vijay Ganesh		2019	SAT+CAS Combination for Mathematics	Assistant Professor, University of Windsor, Canada
PDF, Waterloo	Souraditya Paul	Vijay Ganesh		2014	Cryptanalysis of hash functions	Associate Professor, IIT, Bhilai, India
PhD	Murphy Berzish	Vijay Ganesh		2021	String Solvers	Software Engineer, Amazon
PhD, Waterloo	Saeed Nejati	Vijay Ganesh	Cathy Gebotys	2020	SAT-based Cryptanalysis	Applied Scientist, Amazon, Seattle, USA
PhD, Waterloo	Reza Babaee	Vijay Ganesh	Derek Rayside	2019	ML-based Runtime Verification	Lecturer, University of Waterloo
PhD, Waterloo	Jia Hui Liang	Vijay Ganesh	Krzysztof Czarnecki	2018	Machine Learning for SAT solvers	Senior software engineer at Google

PhD, Waterloo	Ed Zulkoski	Vijay Ganesh	Krzysztof Czarnecki	2018	Complexity-theoretic parameters of SAT formulas	Senior software engineer at Quantstamp
PhD, Waterloo	Curtis Bright	Vijay Ganesh	Krzysztof Czarnecki	2017	SAT+CAS Combination for Mathematics	Postdoctoral Fellow, Waterloo
PhD, Purdue	Yunhui Zheng*	Xiangyu Zhang Purdue U.	Vijay Ganesh	2014	String solvers for web security analysis	Researcher, IBM TJ Watson Center, NY
PhD, Syracuse	Karthick Jayaraman*	Steve Chapin Syracuse U.	Vijay Ganesh	2010	Security analysis of web applications	Software Engineer, Microsoft, Redmond, USA
MASc	Vineel Nagisetty	Vijay Ganesh		2021	Combinations of ML and Logic	Researcher, BorealisAI, Vancouver, Canada
MASc	Laura Graves	Vijay Ganesh		2021	Amnesiac Machine Learning	Software Engineer, Blackberry, Waterloo, Canada
MASc	Behkish Nassirzadeh	Vijay Ganesh		2021	Security of Smart Contracts	PhD student at Waterloo
MASc	Hari Govind	Arie Gurfinkel	Vijay Ganesh	2019	Interpolating Strong Induction	PhD student at Waterloo
MASc	Ian Li	Vijay Ganesh		2019	Complexity of restarts in SAT solvers	PhD student at Waterloo
MASc, Waterloo	Dmitry Blotsky	Vijay Ganesh		2018	Fuzzers for string solvers	Software Engineer at Amazon
MEng, Waterloo	Mark Wheatley	Vijay Ganesh		2018	Symbolic execution-based testing	Software Engineer at BlackBerry
MASc, Waterloo	Riyad Parvez	Vijay Ganesh	Paul Ward	2016	Combining static and symbolic analysis	
MASc, Waterloo	Sanu Subramanian	Vijay Ganesh		2015	String and bit-vector solvers for security	Security Engineer, Intel, Waterloo
MASc, Waterloo	Zack Newsham	Vijay Ganesh	Sebastian Fischmeister	2015	Community structure of Boolean SAT instances	Tech lead, Mushroom Cloud, Toronto
MS, EPFL	Philippe Suter	Viktor Kuncak EPFL	Vijay Ganesh	2008	Combination of decision procedures	Researcher, IBM TJ Watson Center, NY

*I was an unofficial PhD advisor to Yunhui Zheng and Karthick Jayaraman.

SERVICE AND PROFESSIONAL ACTIVITIES

EDITORIAL POSITIONS AND STEERING COMMITTEE MEMBER

- **Editor-in-Chief**
  - Springer Birkhauser Book Series titled "Progress in Computer Science and Applied Logic" (**PCSAL**). Jan 2018-present

- **Associate Editor**
  - Maple Transactions, 2021-present

- **Member of Editorial Board**
  - Journal of Satisfiability (**JSAT**). Oct 2018-present

- **Steering Committee Member**
  - Satisfiability Checking and Symbolic Computation (SC^2) Workshop. 2017-present
  - SAT/SMT/AR Summer School. 2018-present
  - CSTVA Workshop. 2017-present

TECHNICAL PROGRAM COMMITTEES, CHAIR AND REFEREE POSITIONS

Year	Position	Venue
2021	Co-Organizer	1) Simons Institute Semester on "Satisfiability: Theory, Practice, and Beyond", Simons Institute @ Berkeley, USA 2) Waterloo ML + Logic Online Colloquium 3) Dagstuhl Seminar on ML + Logic 4) Theory and Practice of String Solving Workshop 2021
2021	PC Co-chair	ATVA 2021
2021	Senior PC Member	IJCAI 2021 (Senior PC Member)
2021	PC Member	CAV 2021, PLDI 2021, AAAI 2021, FM 2021, SOCS 2021, AAAI-xAI 2021
2020	PC Member	POPL 2020, AAAI 2020, IJCAI 2020, ICTAI 2020
2020	Co-organizer	Dagstuhl seminar: Theory and Practice of SAT Solving (postponed to 2021 due to the COVID pandemic)
2019	PC Member	IJCAI 2019, ICISS 2019, ICTAI 2019
2019	Co-organizer	1) Dagstuhl Seminar: Bringing CP, SAT, and SMT together: Next Challenges in Constraint Solving 2) Waterloo Machine Learning and Security Workshop 3) Waterloo Blockchain and Security Workshop
2018	PC Member	CAV 2018, IJCAI 2018, CP 2018, FM 2018, FCSD 2018, IJCAR 2018, SAT 2018, PoS 2018
2017	PC Member	PLDI 2017 (ERC), ICSE 2017 (NIER), CP 2017, FroCoS 2017, ASL Special Session on Computer-aided Proofs 2017, SYNASC 2017, SC^2 2017

2017	Co-chair	1) Association of Symbolic Logic Special Session on Computer-aided Proofs 2017 2) Satisfiability Checking and Symbolic Computation (SC^2) Workshop 2017 3) David Dill @ 60 Workshop held at CAV 2017
2016	PC Member	POPL 2016 (ERC), ISAIM 2016, ICDCIT 2016, ICCS 2016, FM 2016, PAAR 2016, VSTTE 2016, SC^2 2016, USE 2016
2016	Main Organizer	Fields Institute Workshop on Theoretical Foundations of SAT Solving, Toronto, Canada. Aug 15-19, 2016
2015	PC Member	AAAI 2015, FM 2015, SMT 2015, VSTTE 2015, USE 2015, ASSESS 2015, CCS-SPSM 2015
2015	Co-chair	SMT Workshop 2015
2015	Co-organizer	Dagstuhl seminar: Theory and Practice of SAT Solving, Apr 19-24, 2015
2014	PC Member	ASSESS 2014, CSTVA 2014
2014	Co-chair	Constraint Solvers for Testing, Verification, and Analysis 2014. Held alongside ICSE 2014
2014	Co-organizer	Dagstuhl seminar: Symbolic Execution and Solving, Oct 27-30, 2014
2013	PC Member	FroCos 2013, CSTVA 2013
2012	PC Member	SMT 2012, IEEE NCA 2012
2011	PC Member	SMT 2011, IEEE NCA 2011
2011	Organizer	International Summer School on SAT/SMT Solvers at MIT, Cambridge, MA, USA. June 12-17, 2011
2004	Organizer	International School on Combination of Decision Procedures, SRI International and Stanford, Aug 6-12, 2004

JOURNAL REVIEWER

- International Journal of Artificial Intelligence Tools (IJAIT)
- Mathematical Reviews (MR) – American Mathematical Society
- Communication of the ACM (CACM) – ACM
- Journal of Artificial Intelligence Research (JAIR) – AAAI Press
- Journal of Satisfiability (JSAT)
- Journal of Automated Reasoning (JAR) – Springer
- Artificial Intelligence Journal (AI) – Elsevier
- Formal Methods in Systems Design (FMSD) – Springer
- Algorithms and Complexity in Mathematics, Epistemology, and Science (ACMES)
- ACM Transactions on Software Engineering and Methodology (TOSEM)
- ACM Transactions on Software System and Information Security (TISSEC)

- ACM Transactions on Design Automation of Electronic Systems (TOADES)
- ACM Transactions on Parallel Computing (TOPC)
- ACM Computing Surveys
- IEEE Transactions on Dependable and Securing Computing (TDSC)
- IEEE Transactions on Software Engineering (TSE)
- IEEE Computer Journal
- IEEE Transactions on Reliability
- European Journal of Combinatorics
- Acta Informatica
- AI Communications
- Software Testing, Verification, and Reliability (STVR) – Wiley

INTERNATIONAL REFEREE POSITIONS

- National Science Foundation (NSF), USA, External Reviewer
- European Research Council (ERC), External Reviewer
- Israel Science Foundation (ISF), Israel, External Reviewer
- Royal Swedish Academy of Sciences, External Reviewer
- Icelandic Science Council (IRC), External Reviewer
- Natural Sciences and Engineering Research Council (NSERC), Canada, Reviewer

UNIVERSITY/DEPARTMENTAL COMMITTEES AND INSTITUTES

- University of Waterloo, ECE Department, Graduate Studies Council, 2021 - present
- University of Waterloo, Waterloo AI Institute Co-Director, 2021 - present
- University of Waterloo, Merit Committee, 2021
- University of Waterloo, Engineering Faculty Council, 2020 - present
- University of Waterloo, ECE Department Faculty Hiring Committee Member, 2017 – 2020
- University of Waterloo, Merit Committee, 2020
- University of Waterloo, ECE Department Research Committee, 2017, 2021 - present
- University of Waterloo, Engineering Faculty Council, 2013-2014
- University of Waterloo, ECE Department CGS and Vanier Scholarship Committee, 2014
- University of Waterloo, SE/ECE Capstone Design Symposium Referee, 2014 and 2015

PhD/MS DEFENSE COMMITTEE MEMBER*

Name	Degree	University	Supervisor	Year
Sun Sheng Gu	MASc	Waterloo	Krzysztof Czarnecki	2021
Sima Jamali	PhD	Simon Fraser	David Mitchell	2021
Thibaud Lutillier	PhD	Waterloo	Lin Tan	2020

Mahsa Emam Tabi	PhD	Waterloo	Ladan Tahvildari	2018
Frank Imeson	PhD	Waterloo	Stephen Smith	2017
Vajih Montaghami	PhD	Waterloo	Derek Rayside	2017
Albert Heinle	PhD	Waterloo	Mark Giesbrecht	2016
Ana Sima Claudia	MS	EPFL	George Candea	2015
Elaheh Fata	MASc	Waterloo	Shreyas Sundaram	2013

* Students not under my co-supervision

PhD/MS COMPREHENSIVE EXAM COMMITTEE MEMBER/THESIS READER*

Name	Role	Supervisor	Year
Hari Govind	PhD Comp Exam Committee	Arie Gurfinkel	2020
Thibaud Lutillier	PhD Comp Exam Committee	Lin Tan	2017
Steven Stewart	MS Comp Exam Committee	Derek Rayside	2015
Frank Imeson	PhD Comp Exam Committee	Steven Smith	2015
Mahsa Emam Tabi	PhD Comp Exam Committee	Ladan Tahvildari	2014
Quinn Hamm	MASc Thesis Reader	Lin Tan	2014
Lie Zhang	MASc Thesis Reader	Lin Tan	2014
Vajih Montaghami	PhD Comp Exam Committee	Derek Rayside	2013
Keyvan Golestan	PhD Comp Exam Committee	Fakhri Karray	2013

* Students not under my co-supervision

SOFTWARE DEVELOPED

The primary focus of my research is the theory and practice of mathematical reasoning or constraint solving algorithms (aka SAT and SMT solvers), and in this context, I have led the development of the following solvers and mathematical reasoning: the STP bit-vector and array solver, the HAMPI string solver, the Z3 family of string solvers (Z3str3 and its predecessors Z3str2 and Z3str), the MapleSAT family of Boolean SAT solvers (MapleCOMSPS, MapleSAT etc.), the MaplePainLeSS parallel SAT solver-based cryptanalysis tools, and the MathCheck conjecture verifier.

In addition to constraint solvers, I have also led the development of around dozen software tools such as automatic software testers, compilers, symbolic simulation engines, instruction-set simulators, retargetable compilers for hardware languages, and math conjecture verifiers. Many of my tools are currently in use in more than 100 significant projects in academia and industry.

The MapleSAT Boolean SAT solver is a tool that combines machine-learning based branching and restart heuristics with deduction, a la, conflict analysis. A variant of this solver won two gold medals at the highly competitive SAT competition (main and application tracks) in 2016 and two silver medals at the SAT competition (main and no-limits track) in 2017.

The STP Bitvector and Array Solver is the backend for the KLEE, S2E, BAP, BitBlaze, WebBlaze, and Souper tools. It is used in dozens of companies including at Google, HP, and Apple. The STP solver won the highly competitive annual solver competition, SMTCOMP, in 2006/2010 and placed second in 2011/2014 (bit-vector category). The STP solver played an important role in the development of symbolic execution-based analysis and testing, widely considered a breakthrough in software engineering.

The Z3str family of SMT solvers support a rich theory of strings, regular expressions, and integers aimed at bug-finding, verification, and analysis of string-intensive programs. Additionally, we have also implemented a combination of decision procedure for theories over string equations and bit-vectors in Z3str3 that natively supports functions such as strcpy, strcmp, strncpy, etc. aimed at analysis and testing of C/C++ programs. As of May 2017, Z3str3 string solver is now officially part of Microsoft's Z3 solver.

MathCheck is a tool that combines a SAT solver and a computer algebra system (CAS) aimed at finite verification or counter-example construction for combinatorial math conjectures. As of 2019, we have finitely verified several math conjectures up to bounds not previously attained or have constructed counterexamples for them. For example, using MathCheck we showed that the smallest counterexample to the Williamson conjecture is of order 35, thus settling a problem that had been open since 1944.

SOFTWARE TOOLS

Name of Software	Purpose	Dates	Awards, Conference Venues, and Journals
MapleSAT family of SAT solvers	A SAT solver using a reinforcement learning based branching heuristic	2015 – present	Winner of two silver medals at SAT 2017, and two gold medals at SAT competition 2016. SAT 2017/2016, AAAI 2016, HVC 2015, SAT 2014
STP bit-vector and array solver	A constraint solver for bit-vector and arrays	2005 – present	Winner of SMTCOMP (bit-vector) in 2006/2010, & second in 2011/2014. CAV 2007
Z3 family of String solvers	A solver for string equations, regular expressions, and length	2013 – present	ICSE 2017, FMSD 2017 SMT 2016, CAV 2015, FSE 2013
MathCheck conjecture verifier	A math conjecture verifier via a combination of SAT and CAS	2014 – present	JAR 2017, CASC 2016, Invited paper at IJCAI 2016, CADE 2015
Parallel SAT-based Cryptanalysis	A parallel SAT solver using a ML based splitter	2016 – present	SAT 2017, CP 2018
MPro	Smart contract security analysis tool combining static & symbolic analysis	2019 – present	ISSRE 2019

Lynx	A programmatic Boolean SAT solver	2011 – 2012	SAT 2012
HAMPI	A solver for string equations, and regular expressions	2009 – 2012	CAV 2011, ACM SIGSOFT Award at ISSTA 2009 (also, an invited paper at TOSEM 2012)
Mohawk	A model-checking tool for access-control policies	2010 – 2011	CCS 2011, TISSEC 2013
BuzzFuzz	A taint-based whitebox fuzzer for C programs	2008 – 2009	ICSE 2009
jFuzz	A concolic testing tool for Java	2009	NFM 2009

REFERENCES

Professor Moshe Vardi
Department of Computer Science
Mail Stop 132
Rice University, 6100 S. Main Street
Houston, TX, USA 77005-1892
Phone: +1-713-348-5977
Email: vardi@cs.rice.edu

Professor Martin Rinard
Department of Computer Science
Massachusetts Institute of Technology, 32-G828
32 Vassar Street
Cambridge, MA, USA 02139
Phone: +1-617-258-6922
Email: rinard@csail.mit.edu

Professor Somesh Jha
Computer Sciences Department
University of Wisconsin, Madison
1210 W. Dayton Street
Madison, WI, USA 53706
Phone: +1-608-262-9519
Email: jha@cs.wisc.edu

Professor Sam Buss
Department of Mathematics
University of California, San Diego
Office APM 7456
La Jolla, CA, USA 92093-0112
Phone: +1-858-534-6455
Email: sbuss@ucsd.edu

Professor Ras Bodik
Paul G. Allen School of Computer Science
and Engineering
University of Washington, Seattle
Bill and Melinda Gates Center, Room 243
3000 E Stevens Way NE
Seattle, WA, USA 98195
Phone: +1-206-616-7172
Email: bodik@cs.washington.edu