Joel Galenson

Contact Information

650-804-6870
jgalenson@gmail.com
http://jgalenson.github.io/

Research Interests

Program synthesis, static and dynamic analysis, testing, compilers, language design
AI, LLMs, data science, prompt engineering
Systems security, return-oriented programming, exploit development, malware detection

Education

University of California, Berkeley 2014
Ph.D.
Advisors: Rastislav Bodik and Koushik Sen

Stanford University 2008
B.S. (honors, distinction)

Honors and Awards

2^{nd} place, LIVE 2013 2013
Best Student Paper Award, ADKDD 08 2008
Winner of 3D video game competition at Stanford 2007
Tau Beta Pi (junior year) 2007

Publications

Joel Galenson, Cindy Rubio-González, Sarah Chasins, and Liang Gong. Research.js: Evaluating Research Tool Usability on the Web. In *Proceedings of the 5th Workshop on Evaluation and Usability of Programming Languages and Tools (PLATEAU 2014)*, Portland, Oregon, USA, 2014.

Joel Galenson. Dynamic and Interactive Synthesis of Code Snippets. Ph.D. Dissertation, 2014.

Joel Galenson, Philip Reames, Rastislav Bodik, Bjoern Hartmann, and Koushik Sen. CodeHint: Dynamic and Interactive Synthesis of Code Snippets. In *International Conference on Software Engineering (ICSE 2014)*, Hyderabad, India, 2014.

Mihai Budiu, Joel Galenson, and Gordon D. Plotkin. The Compiler Forest. In *Proceedings of the 22nd European conference on Programming Languages and Systems (ESOP 2013)*, Rome, Italy, 2013.

David Gay, Joel Galenson, Mayur Naik, and Kathy Yelick. Yada: Straightforward Parallel Programming. In *Parallel Computing*, Elsevier, 2011.

Rastislav Bodik, Satish Chandra, Joel Galenson, Doug Kimmelman, Nicholas Tung, Shaon Barman, and Casey Rodarmor. Programming with Angelic Nondeterminism. In *Proceedings of the 37th Symposium on Principles of Programming Languages (POPL 2010)*, Madrid, Spain, 2010.

Jason Auerbach, Joel Galenson, and Mukund Sundararajan. An empirical analysis of return on investment maximization in sponsored search auctions. In *Proceedings of the Second International Workshop on Data Mining and Audience Intelligence for Advertising (ADKDD 2008)*, Las Vegas, Nevada, USA, 2008.

Refereed Presentations

CodeHint: Dynamic and Interactive Synthesis for Modern IDEs. Future Programming Workshop, SPLASH, 2014.

CodeHint: Dynamic and Interactive Synthesis for Modern IDEs. Future Programming Workshop, Strange Loop, 2014.

Code Hint. First International Workshop on Live Programming, 2013.

Experience

Software Engineer, Google/X, the moonshot factory — Winter 2022 - Present

I am working on program synthesis with AI. I have worked on products that launched in Google Search, Colab, Android Studio, Google Cloud, and more. I have focused on areas including improving model correctness, training classifiers, and building the products themselves.

Software Engineer, Google — Spring 2017 - Winter 2022

I was on the Android Platform Security team, where I have worked on multiple projects to improve the security and privacy of billions of Android devices.

- I worked on adding support for the Rust language to the Android platform to improve its security, which includes writing Rust code, working with upstream, and integrating it into the Android platform.
- I proposed, co-designed, and implemented one of Android's premier upcoming privacy features, the Privacy dashboard.
- I have made numerous optimizations to the upstream LLVM compiler to improve the performance of components such as its integer overflow sanitizer and undefined behavior sanitizer. This work was crucial in allowing us to enable these features on production devices.
- I was one of the maintainers of Android's SELinux system, which uses mandatory access controls to improve the security of the system. I have worked on improving Android's core policy and on bringing up multiple new devices.

Senior Engineer, Qualcomm Research Silicon Valley — Fall 2014 - Spring 2017

- I researched behavioral mobile security solutions to protect against malware and exploits. I have spent much of my time developing attacks on Android, including building real exploits that bypass SELinux and target Chrome and the Stagefright and Dirtycow bugs. I have handwritten ARM assembly and built a simple shellcode and ROP compiler to ease payload development. I developed and gave our lab a tutorial on memory error attacks and defenses, including building a sequence of ROP attacks from simple to complex.
- I worked on developing compilation techniques for programming special purpose accelerator architectures. Our compiler was based on LLVM, and I worked on the backend, including scheduling, software pipelining, optimizing individual instructions, co-designing new instructions, and numerous architecture-specific passes. I also worked on providing tools to understand and optimize the compiler output as well as improving our test infrastructure and tracking upstream development.

Graduate Student Researcher, University of California, Berkeley — Fall 2008 - Summer 2014

I was a member of the Parallel Computing Laboratory (Par Lab) where I worked on program synthesis techniques to aid general-purpose programming. I built an Eclipse plugin that dynamically generated code snippets (along with a JavaScript port) and a graphical programming by demonstration tool.

Teaching Assistant, University of California, Berkeley — Spring 2014

Was a TA for CS 61B: Data Structures.

Intern, Microsoft Research Silicon Valley — Summer 2011

Worked on an architecture for modular cooperating compilers.

Intern, Microsoft Research Silicon Valley — Summer 2010

Worked on a new architecture for evaluating LINQ queries that encompasses DryadLINQ.

Teaching Assistant, University of California, Berkeley — Fall 2009
Was a TA for CS 164: Programming Languages and Compilers.

Intern, IBM T.J. Watson Research Center — Summer 2009
Worked on the constraint-based type system for the X10 language.

Platform intern, Mozilla — Summer 2008
Worked on a native code compiler for regular expressions.

Section Leader for CS 106, Stanford University — Fall 2005 - Summer 2008
Taught a section covering introductory programming topics, graded homework and exams, staffed a help desk.

Researcher, Stanford University — Summer 2006 - Spring 2008

- Built a verifying compiler for Zohar Manna and Aaron Bradley.
- Worked on two static analysis tools for Zohar Manna.
- Investigated the properties of online ad auctions and bidder strategies with Tim Roughgarden.
- Developed methods to enable the use of remote computers to speed up data processing by a robot for Andrew Ng.
- Developed techniques for visualizing personal information spaces for Pat Hanrahan.

Teaching Assistant, Stanford University — Winter 2008
Was a TA for CS 156: Calculus of Computation.

Resident Computer Consultant, Stanford University — Fall 2006 - Spring 2008
Assisted undergraduates with personal computer problems and administered a dorm network.

Professional Activities

Artifact Evaluation Committee: POPL	2015
External reviewer: PLDI, CAV	2014
External reviewer: ASPLOS, OOPSLA, VMCAI	2013
Graduate Admissions Committee, UC Berkeley	2009

Leadership

Computer Science Graduate Student Association member	Fall 2013 - Spring 2014
Graduate Assembly committee representative	Fall 2013 - Spring 2014
Organized UC Berkeley Programming Languages seminars	Fall 2009 - Summer 2014

Computer Skills

C++, Java, Rust, Python, Go, Scala, OCaml, C#, JavaScript, C, ARM, LaTeX, HTML
Linux, Android, return-oriented programming

References

Available on request