

Robert Dockins

Galois, Inc.
421 SW 6th Ave, Suite 300
Portland, OR 97204

robdockins@fastmail.fm
<http://rwd.rdockins.name>

Work History

2014–present Software Engineer. Galois, Inc. Portland, OR.
2012–2014 Postdoctoral research. Portland State University. Portland, OR.
2009 (Summer) Research Co-op. Rockwell-Collins. Cedar Rapids, IA.
2005–2007 Digital Resources Archivist. DCA, Tufts University. Medford, MA.
2004–2005 Project Analyst. DCA, Tufts University. Medford, MA.

Education

2012 Ph.D., Computer Science. Princeton University.
2007 M.S., Computer Science. Tufts University.
2003 B.S., Computer Science. Summa Cum Laude. Baylor University.

Awards

2011 Siebel Scholar, Class of 2012.
2007 First-year fellowship, Princeton University.
2003 Outstanding Computer Science Scholar, Baylor University.
1999 National Merit Scholarship Finalist.

Teaching Experience

- Spring 2011. Teaching assistant, COS333: Advanced Programming Techniques.
- Spring 2009. Preceptor for 2 sections of COS126: General Computer Science.
- Fall 2008. Teaching assistant, COS441: Programming Languages.

Research Interests

- Programming Languages: declarative programming, type systems, semantics, compiler correctness.
- Formal Logic: type theory, theorem proving, constructive mathematics, PL metatheory.
- Concurrency: concurrent separation logic, verification of concurrent programs.

Research Experience

The main thrust of my postdoctoral current research occurred under the DARPA MRC/SOUND (Mission-oriented Resilient Clouds / Safety On Untrusted Network Devices) research project. Together with Andrew Tolmach, I worked on developing a domain-specific programming language for expressing policy. The language aims to provide a better way to express desired policy behavior than existing options. Furthermore, the language is designed specifically to be amenable to various kinds of static analysis and formal proof, such as termination analysis and policy conflict detection.

I am also currently pursuing research goals related to the constructive formalization of domain theory in Coq, and also on finding better ways to use separation logic to reason about the correctness of imperative programs that manipulate pointer-graphs.

During my PhD studies at Princeton, I worked with my adviser, Andrew Appel, on developing multimodal substructural logics specifically designed to be used for reasoning about programming languages in the proof assistant Coq. The goal of this research was to develop new tools for foundational reasoning about the correctness of compilers and the soundness of axiomatic semantics of programming languages. My PhD dissertation was on the topic of compiler correctness: specifically, I investigated improvements to the correctness proof for the CompCert verified C compiler. All the proofs and constructions referenced in the dissertation were formalized using the proof assistant Coq.

The primary focus of my Master's research involved developing a system for verifying the type-safety of Haskell programs expressed as G-Machine bytecode. Two major technical innovations in this work are an explicitly-typed variant of System F_ω to be used as an intermediate language for typed Haskell compilation, and algorithms for generating and checking type certificates for compiled bytecode programs.

In 2004 and 2005, I participated in a research project conducted jointly by Tufts University and Yale University regarding the preservation of electronic records (NHPRC grant number 2004-083). My duties involved assisting the co-principal investigators to

research aspects of the project that dealt with computer software and writing proof-of-concept software to test the methodologies developed during the project. I also helped draft and edit the project findings.

Academic Service

- Workshop on Syntax and Semantics of Low-Level Languages (LOLA) 2012: Program committee member

Papers and Presentations

- “Formalized, Effective Domain Theory in Coq.” Robert Dockins. To appear in *Interactive Theorem Proving (ITP)* 2014.
- **Program Logics for Certified Compilers.** Andrew W. Appel with Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds, Gordon Stewart, Sandrine Blazy and Xavier Leroy. Cambridge University Press, 2014.
- “Verified Compilation for Shared-memory C.” Lennart Beringer, Gordon Stewart, Robert Dockins and Andrew W. Appel. To appear in *European Symposium on Programming (ESOP)* 2014.
- “A List-machine Benchmark for Mechanized Metatheory.” Andrew W. Appel, Robert Dockins and Xavier Leroy. *Journal of Automated Reasoning*. Volume 49, issue 3. Pages 453–491. October 2012.
- *Operational Refinement for Compiler Correctness*. Doctoral Dissertation, Princeton University. September 2012.
- “Time Bounds for General Function Pointers.” Robert Dockins and Aquinas Hobor. *The 28th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXVIII)*. ENTCS 286. Pages 139–155. June 2012.
- “A Logical Mix of Approximation and Separation.” Aquinas Hobor and Robert Dockins. In Proc. *8th Asian Symposium on Programming Languages and Systems (APLAS)*. LNCS 6461. Pages 439–454. November 2010. Invited tutorial with paper.
- “A Theory of Indirection via Approximation.” Aquinas Hobor, Robert Dockins and Andrew W. Appel. In Proc. *37th Annual ACM SIGACT-SIGPLAN Symposium on Principles in Programming Languages (POPL)*. ACM SIGPLAN Notices, Volume 45, Issue 1. Pages 171–184. January 2010.

- “A Fresh Look at Separation Algebras and Share Accounting.” Robert Dockins, Aquinas Hobor and Andrew W. Appel. In Proc. *7th Asian Symposium on Programming Languages and Systems (APLAS)*. LNCS 5904. Pages 161–177. December 2009.
- “Comparing Semantic and Syntactic Methods in Mechanized Proof Frameworks.” C.J. Bell, Robert Dockins, Aquinas Hobor, Andrew W. Appel and David Walker. Workshop paper. In the *2nd International Workshop on Proof-Carrying Code (PCC)*. June 2008.
- “Multimodal Separation Logic for Reasoning about Operational Semantics.” Robert Dockins, Andrew Appel, and Aquinas Hobor. In Proc. *Mathematical Foundations of Programming Semantics (MFPS) XXIV*. ENTCS, Volume 218. Pages 5–20. October 2008. Elsevier.
- “Bytecode Verification for Haskell.” Robert Dockins and Samuel Z. Guyer. Workshop Paper. *The Symposium on Trends in Functional Programming (TFP)* 2007.
- “Bytecode Verification for Haskell.” Robert Dockins and Samuel Z. Guyer. Technical Report, Tufts University Dept. of Computer Science. TR-2007-2. 2007.
- “The Ingest and Maintenance of Electronic Records: Moving from Theory to Practice.” Kevin L. Glick, Eliot Wilczek, Robert Dockins. Poster Presentation. *Joint Conference on Digital Libraries (JCDL)* 2006.