

Michael D. Adams

School of Computing, University of Utah
50 S. Central Campus Drive
Room 3190
Salt Lake City, UT 84112

<http://michaeldadams.org>
+1-801-581-5843 (Fax)

Research Interests

Programming Languages; Cybersecurity; Next-Generation Languages; Static Analysis; Parsing; Compilation; Optimization; Meta-programming and Macros; Generic Programming; Type Systems

Academic Degrees

Indiana University

Doctor of Philosophy in Computer Science
Minor in Logic
Advisor: R. Kent Dybvig

Bloomington, Indiana

October 2011

University of Kansas

Bachelor of Science in Computer Science
Bachelor of Science in Computer Engineering
Minor in Mathematics
Honors and Highest Distinction

Lawrence, Kansas

May 2005

Employment History

University of Utah

Research Assistant Professor
Postdoctoral Research Associate

Salt Lake City, Utah

January 2016 – Present

August 2014 – December 2015

University of Illinois at Urbana-Champaign

Postdoctoral Research Associate

Urbana, Illinois

July 2013 – July 2014

Portland State University

Postdoctoral Research Associate

Portland, Oregon

August 2011 – June 2013

Funding

- **PI.** DARPA BAA-14-60: Space/Time Analysis for Cybersecurity (STAC). “Automated Analysis of Algorithmic Attack Vulnerabilities”: **\$3M (2015–2019)**. AFRL FA8750-15-2-0092.
- **PI.** DARPA Seedling. “Extensively Tested Java Decompilation”: **\$0.3M (2019–2021)**.
- **Co-PI.** DARPA BAA HR001118S0054: Safe Documents (SafeDocs). “SPRACO: Securing PDFium with Rust and Cross-Origin Data-Flow Integrity”: **\$6M (2019–2023)**. (Submitted, awaiting review)

Publications

Refereed Journals and Conferences

- Kimball Germane, Jay McCarthy, Michael D. Adams, and Matthew Might. *Demand Control-Flow Analysis*. In Proceedings of the 20th International Conference on Verification, Model Checking, and Abstract Interpretation, **VMCAI '19**. (Submitted, awaiting review)
- Thomas Gilray, Michael D. Adams, and Matthew Might. *Abstract allocation as a unified approach to polyvariance in control-flow analyses*. **Journal of Functional Programming**, 28:E18, August **2018**. doi:10.1017/S0956796818000138
- Michael D. Adams and Matthew Might. *Restricting grammars with tree automata*. Proceedings of the ACM on Programming Languages, 1(**OOPSLA '17**):82:1–82:25, October **2017**. ISSN 2475-1421. doi: 10.1145/3133906.
- William Mansky, Elsa L. Gunter, Dennis Griffith, and Michael D. Adams. *Specifying and executing optimizations for generalized control flow graphs*. **Science of Computer Programming**, 130:2–23, November **2016**. ISSN 0167-6423. doi: 10.1016/j.scico.2016.06.003.
- Thomas Gilray, Michael D. Adams, and Matthew Might. *Allocation characterizes polyvariance: A unified methodology for polyvariant control-flow analysis*. In Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, **ICFP '16**, pages 407–420. ACM, New York, NY, USA, September 2016. ISBN 978-1-4503-4219-3. doi: 10.1145/2951913.2951936.
- Michael D. Adams, Celeste Hollenbeck, and Matthew Might. *On the complexity and performance of parsing with derivatives*. In Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation, **PLDI '16**. ACM, New York, NY, USA, June 2016. doi: 10.1145/2908080.2908128.
- Thomas Gilray, Steven Lyde, Michael D. Adams, Matthew Might, and David Van Horn. *Push-down control-flow analysis for free*. In Proceedings of the 43rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, **POPL '16**. ACM, New York, NY, USA, January 2016. doi: 10.1145/2837614.2837631.
- Michael D. Adams, Andrew Farmer, and José Pedro Magalhães. *Optimizing SYB traversals is easy!*. **Science of Computer Programming**, 112, Part 2:170–193, November **2015**. ISSN 0167-6423. doi: 10.1016/j.scico.2015.09.003.
- Michael D. Adams. *Towards the Essence of Hygiene*. In Proceedings of the 42nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, **POPL '15**. ACM, New York, NY, USA, 2015. doi: 10.1145/2676726.2677013.
- Michael D. Adams. *Principled parsing for indentation-sensitive languages: Revisiting Landin's offside rule*. In Proceedings of the 40th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages, **POPL '13**, pages 511–522. ACM, New York, NY, USA, 2013. doi: 10.1145/2429069.2429129.
- Michael D. Adams, Andrew W. Keep, Jan Midtgaard, Matthew Might, Arun Chauhan, and R. Kent Dybvig. *Flow-sensitive type recovery in linear-log time*. In Proceedings of the 2011 ACM International Conference on Object Oriented Programming Systems Languages and Applications, **OOPSLA '11**, pages 483–498. ACM, New York, NY, USA, October 2011. ISBN 978-1-4503-0940-0. doi: 10.1145/2048066.2048105.

- Michael D. Adams and R. Kent Dybvig. *Efficient nondestructive equality checking for trees and graphs*. In Proceeding of the 13th ACM SIGPLAN international conference on Functional programming, **ICFP '08**, pages 179–188. ACM, New York, NY, USA, 2008. doi: 10.1145/1411204.1411230.
- Peter Gottschling, David S. Wise, and Michael D. Adams. *Representation-transparent matrix algorithms with scalable performance*. In Proceedings of the 21st annual international conference on Supercomputing, **ICS '07**, pages 116–125. ACM, New York, NY, USA, 2007. doi: 10.1145/1274971.1274989.

Refereed Symposia and Workshops

- Michael D. Adams and Ömer S. Ağacan. *Indentation-sensitive Parsing for Parsec*. In Proceedings of the 2014 ACM SIGPLAN Symposium on Haskell, **Haskell '14**, pages 121–132. ACM, New York, NY, USA, 2014. doi: 10.1145/2633357.2633369.
- Michael D. Adams, Andrew Farmer, and José Pedro Magalhães. *Optimizing SYB is easy!*. In Proceedings of the ACM SIGPLAN 2014 Workshop on Partial Evaluation and Program Manipulation, **PEPM '14**, pages 71–82. ACM, New York, NY, USA, 2014. doi: 10.1145/2543728.2543730. (Received the **PEPM '14 Best Paper Award**.)
- Michael D. Adams and Thomas M. DuBuisson. *Template your boilerplate: Using Template Haskell for efficient generic programming*. In Proceedings of the 2012 ACM SIGPLAN Haskell symposium, **Haskell '12**, pages 13–24. ACM, New York, NY, USA, 2012. doi: 10.1145/2364506.2364509.
- Jan Midtgaard, Michael Adams, and Matthew Might. *A structural soundness proof for Shivers's escape technique: A case for Galois connections*. In Antoine Miné and David Schmidt, editors, Static Analysis, **SAS '12**, volume 7460 of Lecture Notes in Computer Science, pages 352–369. Springer Berlin / Heidelberg, 2012. doi: 10.1007/978-3-642-33125-1_24.
- Michael D. Adams. *Scrap your zippers: A generic zipper for heterogeneous types*. In Proceedings of the 2010 ACM SIGPLAN workshop on Generic programming, **WGP '10**, pages 13–24. ACM, New York, NY, USA, 2010. doi: 10.1145/1863495.1863499.
- Andrew W. Keep, Michael D. Adams, Lindsey Kuper, William E. Byrd, and Daniel P. Friedman. *A pattern matcher for miniKanren or how to get into trouble with CPS macros*. In Proceedings of the 2009 Scheme and Functional Programming Workshop, **Scheme '09**, number CPSLO-CSC-09-03 in California Polytechnic State University Technical Report, pages 37–45. 2009. URL http://digitalcommons.calpoly.edu/csse_fac/83/.
- Michael D. Adams and David S. Wise. *Seven at one stroke: Results from a cache-oblivious paradigm for scalable matrix algorithms*. In Proceedings of the 2006 workshop on Memory system performance and correctness, **MSPC '06**, pages 41–50. ACM, New York, NY, USA, 2006. doi: 10.1145/1178597.1178604.

Unrefereed

- Michael D. Adams and Matthew Might. *Disambiguating grammars with tree automata*. In Proceedings of **Parsing@SLE**. ACM, New York, NY, USA, October 2015.
- Michael D. Adams. *Flow-Sensitive Control-Flow Analysis in Linear-Log Time*. **Ph.D. Thesis**, Indiana University, 2011.
- Michael D. Adams and David S. Wise. *Fast additions on masked integers*. **SIGPLAN Notices**, 41(5):39–45, May 2006. ISSN 0362-1340. doi: 10.1145/1149982.1149987.
- Michael D. Adams. *The representation of constraints, annotations and first class patterns over arbitrary data types in Haskell*. **Honors Undergraduate Research**, University of Kansas, May 2004.

Teaching Experience

University of Illinois at Urbana/Champaign

Undergraduate Research Opportunities in Computing

Urbana, Illinois*April 2014 – July 2014*

- Research Mentor for four undergraduate students

Indiana University

Associate Instructor

Bloomington, Indiana*January 2009 – May 2011*

- **Associate Instructor of the Year (2009–2010)** – Computer Science
- CSCI H212: Introduction to Software Systems, Honors: Spring 2011
- CSCI H211: Introduction to Computer Science, Honors: Fall 2010
- CSCI C343/A594: Data Structures: Fall 2009 and Spring 2010
- CSCI C212/A592: Introduction to Software Systems: Spring 2009

Undergraduate Research Opportunities in Computing

January 2011 – May 2011

- Research Mentor for two undergraduate students

Software

I have been involved in the development of a number of languages and compilers, including

- the **Glasgow Haskell Compiler**,
- the **Chez Scheme** compiler,
- the **X10** language,
- the **Habit** compiler,
- the **Hermit** optimization system, and
- the **K Framework**.

I am the principal developer of a number of open source libraries and tools:

- **Jade: The Extensively Tested Java Decompiler** (JVM bytecode to Java decompiler)
<http://github.com/Ucombinator/jade>
- **Jaam: JVM Abstracting Abstract Machine** (Static analysis tool for JVM bytecode)
<http://github.com/Ucombinator/jaam>
- **tree-automata** (Regular tree-automata library)
<https://github.com/svenkeidel/tree-automata>
(Though I was the original developer, maintenance has been handed over to Sven Keidel.)
- **Derp 3** (Parsing library)
<https://bitbucket.org/ucombinator/derp-3>
- **Hermit SYB** (Optimizer for SYB (Scrap Your Boilerplate) code)
<https://github.com/xich/hermit-syb/>

- **indentation-parsec** (Parsec parser extension for indentation)
<https://hackage.haskell.org/package/indentation-parsec>
- **indentation-trifecta** (Trifecta parser extension for indentation)
<https://hackage.haskell.org/package/indentation-trifecta>
- **Template Your Boilerplate** (Generic programming library)
<https://hackage.haskell.org/package/TYB>
- **Scrap Your Zippers** (Generic zipper library)
<https://hackage.haskell.org/package/syz>

Service

- International Conference on Functional Programming (**ICFP**) 2019
External Review Committee <https://icfp19.sigplan.org/>
- Symposium on Principles of Programming Languages (**POPL**) 2019
Program Committee <https://popl19.sigplan.org/>
- International Conference on Functional Programming (**ICFP**) 2018
External Review Committee <https://icfp18.sigplan.org/>
- Haskell Symposium (**Haskell**) 2018
Program Committee <https://www.haskell.org/haskell-symposium/2018/>
- Principles and Practice of Declarative Programming (**PPDP**) 2015
Program Committee <http://costa.ls.fi.upm.es/ppdp15/>
- Scheme Workshop (**Scheme**) 2014
Program Committee <http://homes.soic.indiana.edu/jhemann/scheme-14/>
- Principles and Practice of Declarative Programming (**PPDP**) 2014
Program Committee <http://users-cs.au.dk/danvy/ppdp14/>
- Scheme Workshop (**Scheme**) 2012
Program Committee <http://users-cs.au.dk/danvy/sfp12/>
- Scheme Workshop (**Scheme**) 2011
Program Committee <http://scheme2011.ucombinator.org/>