Michael D. Adams

National University of Singapore
Singapore, Singapore
michaeldadams.org

Research Interests

Programming Languages; Software Engineering; Software Security; Static Analysis; Compilation; Optimization; Meta-programming

Education

2011 **Ph.D., Computer Science** with Minor in Logic, *Indiana University*, Bloomington, IN
Dissertation: Flow-Sensitive Control-Flow Analysis in Linear-Log Time
Advisor: R. Kent Dybvig

2005 **B.S., Computer Science** and **B.S., Computer Engineering** with Minor in Math (Highest Distinction), *University of Kansas*, Lawrence, KS
Honors thesis: The representation of constraints, annotations and first class patterns over arbitrary data types in Haskell.
Honors Advisor: Perry Alexander

Employment History

2022–Present **Assistant Professor**, *National University of Singapore*, Singapore

2021–2025 **Assistant Professor**, *Yale-NUS College*, Singapore

2019–2021 **Assistant Research Scientist**, *University of Michigan*, Ann Arbor, MI, USA

2016–2019 **Research Assistant Professor**, *University of Utah*, Salt Lake City, UT, USA

2014–2015 **Postdoctoral Research Associate**, *University of Utah*, Salt Lake City, UT, USA

2013–2014 **Postdoctoral Research Associate**, *University of Illinois at Urbana-Champaign*, Urbana, IL,USA

2011–2013 **Postdoctoral Research Associate**, *Portland State University*, Portland, OR, USA

Publications

Refereed Journals

ICFP '25 Edward Peters, Yong Qi Foo, and Michael D. Adams. Pushing the Information-Theoretic Limits of Random Access Lists: Traversing Cons Lists in$(1 + 1/\sigma)\ \lfloor \lg n \rfloor + \sigma + 9$ Steps. *Proceedings of the ACM on Programming Languages*, 9(ICFP '25), October 2025.

POPL '25 Michael D Adams, Eric Griffis, Thomas J Porter, Sundara Vishnu Satish, Eric Zhao, and Cyrus Omar. Grove: A Bidirectionally Typed Collaborative Structure Editor Calculus. *Proceedings of the ACM on Programming Languages*, 9(POPL '25), January 2025.

ICFP '20 Pierce Darragh and Michael D. Adams. Parsing with Zippers. *Proceedings of the ACM on Programming Languages*, 4(ICFP '20), August 2020.

JFP '18 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.

OOPSLA '17 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.

SCP '16 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.

SCP '15 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.

Refereed Conferences

ESOP '20 Kimball Germane, Michael D. Adams. Liberate Abstract Garbage Collection from the Stack by Decomposing the Heap. In *Programming Languages and Systems - 29th European Symposium on Programming*, pages 197–223. Springer, April 2020.

VMCAI '19 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*, January 2019.

ICFP '16 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*, pages 407–420. ACM, New York, NY, USA, September 2016.

PLDI '16 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*. ACM, New York, NY, USA, June 2016.

POPL '16 Thomas Gilray, Steven Lyde, Michael D. Adams, Matthew Might, and David Van Horn. Pushdown control-flow analysis for free. In *Proceedings of the 43nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. ACM, New York, NY, USA, January 2016.

POPL '15 Michael D. Adams. Towards the Essence of Hygiene. In *Proceedings of the 42nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. ACM, New York, NY, USA, January 2015.

POPL '13 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*, pages 511–522. ACM, New York, NY, USA, 2013.

OOPSLA '11 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*, pages 483–498. ACM, New York, NY, USA, October 2011.

ICFP '08 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*, pages 179–188. ACM, New York, NY, USA, 2008.

ICS '07 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*, pages 116–125. ACM, New York, NY, USA, 2007.

Refereed Symposia and Workshops

PEPM '14 **Best Paper Award**
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*, pages 71–82. ACM, New York, NY, USA, 2014.

Haskell '14
Michael D. Adams and Ömer S. Ağacan. Indentation-sensitive Parsing for Parsec. In *Proceedings of the 2014 ACM SIGPLAN Symposium on Haskell*, pages 121–132. ACM, New York, NY, USA, 2014.

Haskell '12
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*, pages 13–24. ACM, New York, NY, USA, 2012.

SAS '12
Jan Midtgaard, Michael Adams, and Matthew Might. A structural soundness proof for Shivers's escape technique: A case for Galois connections. In *Static Analysis*, volume 7460 of Lecture Notes in Computer Science, pages 352–369. Springer Berlin / Heidelberg, 2012.

WGP '10
Michael D. Adams. Scrap your zippers: A generic zipper for heterogeneous types. In *Proceedings of the 2010 ACM SIGPLAN workshop on Generic programming*, pages 13–24. ACM, New York, NY, USA, 2010.

Scheme '09
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*, number CPSLO-CSC-09-03 in California Polytechnic State University Technical Report, pages 37–45. 2009. URL `http://digitalcommons.calpoly.edu/csse_fac/83/`.

MSPC '06
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*, pages 41–50.

Other

Whitepaper
Yong Qi Foo, Brian Sze-Kai Cheong, Michael D. Adams. Fixed-Point-Oriented Programming: A Concise and Elegant Paradigm. arXiv:2507.21439, July 2025.

Parsing@SLE '15
Michael D. Adams and Matthew Might. Disambiguating grammars with tree automata. In Proceedings of Parsing@SLE. ACM, New York, NY, USA, October 2015.

Ph.D. Thesis '11
Michael D. Adams. Flow-Sensitive Control-Flow Analysis in Linear-Log Time. *Ph.D. Thesis*, Indiana University, 2011.

SIGPLAN Notices '06
Michael D. Adams and David S. Wise. Fast additions on masked integers. *SIGPLAN Notices*, 41(5):39–45, May 2006.

Honors Undergraduate Research '04
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.

Software

I have been involved developing of a number of languages and compilers, including:

- **Glasgow Haskell Compiler (GHC)**
  `https://www.haskell.org/ghc/`
- **Chez Scheme**
  `https://scheme.com/`
- **X10**
  `http://x10-lang.org/`
- **Habit**
  `https://www.habit-lang.org/`

- **Hermit**
  `https://ku-fpg.github.io/software/hermit/`
- **K Framework**
  `https://kframework.org/`
- **Hazel**
  `https://hazel.org/`

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

- **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, Sven Keidel now handles its maintenance.
- **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

- Conference on Programming Language Design and Implementation (**PLDI**) 2026
  Program Committee `https://pldi26.sigplan.org/`
- miniKanren 2025 (**miniKanren 2025**) 2025
  Program Chair `https://conf.researchr.org/home/icfp-splash-2025/minikanren-2025`
- ML Family Workshop 2025 (**ML**) 2025
  Program Committee `https://conf.researchr.org/home/icfp-splash-2025/mlsymposium-2025`
- Haskell Symposium (**Haskell**) 2024
  Program Committee `https://www.haskell.org/haskell-symposium/2024/`
- International Conference on Functional Programming (**ICFP**) 2023
  Program Committee `https://icfp23.sigplan.org/`

- European Symposium on Programming (**ESOP**) 2022
  Program Committee `https://etaps.org/2022/esop`
- Symposium on Principles of Programming Languages (**POPL**) 2022
  Program Committee `https://popl22.sigplan.org/`
- Scheme Workshop (**Scheme**) 2020
  Program Chair `https://icfp20.sigplan.org/home/scheme-2020`
- Scheme Workshop (**Scheme**) 2019
  Program Committee `https://thomas.gilray.org/scheme-2019/`
- International Conference on Functional Programming (**ICFP**) 2019
  External Review Committee `https://icfp19.sigplan.org/`
- SPLASH Student Research Competition (**SPLASH SRC**) 2019
  Reviewer `https://2019.splashcon.org/track/splash-2019-SRC`
- 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/`

(Document last updated July 2025)