# Alex D. Groce

### Professional Preparation

North Carolina State University Raleigh, NC Computer Science B.S. 1999

North Carolina State University Raleigh, NC Multidisciplinary Studies B.S. 1999

Carnegie Mellon University Pittsburgh, PA Computer Science Ph.D. 2005

### Appointments

Jan 2017- Associate Professor with indefinite tenure, School of Informatics,

Computing, and Cyber Systems, Northern Arizona University

Feb 2019- Security Engineer (sabbatical), Trail of Bits

Aug 2019

2015-2017 Associate Professor with indefinite tenure, School of Electrical Engineering and Computer Science, Oregon State University

2009-2015 Assistant Professor, School of Electrical Engineering and Computer

Science, Oregon State University

2008, 2009 Lecturer in Computer Science, Part-Time, California Institute of

Technology

2005-2009 Core Member, Laboratory for Reliable Software, Jet Propulsion

Laboratory

1999-2005 Graduate Research Assistant, Computer Science Department, Carnegie Mellon University

2001, 2002 Summer Research Intern, NASA Ames Research Center

### 2000 Summer Research Intern, Lucent/Bell Laboratories Innovations, Murray

### Hill

1999 Summer Research Intern, Computer Science Department, SUNY Stony Brook

1998 Summer Research Intern, Computer Science Department, North Carolina State University

### Publications

### Relevant Publications

1. **Groce A**, Holmes J, Marinov D, Shi A, Zhang L. An Extensible, Regular-Expression Based Tool for Multi-Language Mutant Generation. ACM/IEEE International Conference on Software Engineering, 25-28, May-June 2018.
2. Gopinath R, Alipour A, Ahmed I, Jensen C, **Groce A**. On the limits of mutation reduction strategies. ACM/IEEE International Conference on Software Engineering, 511-522, May 2016.
3. **Groce A**, Ahmed I, Jensen C, McKenney P.How Verified is my Code? Falsification-Driven Verification. IEEE/ACM International Conference on Software Engineering, 737-748, November 2015.
4. Gopinath R, Alipour A, Ahmed I, Jensen C, **Groce A**. How hard does mutation testing have to be, anyway? IEEE International Symposium on Software Reliability Engineering, 216-227, November 2015.
5. Gopinath R, Jensen C, **Groce A**. Mutations: How Close are they to Real Faults? IEEE International Symposium on Software Reliability Engineering, 189-200, November 2014.

### Other Significant Publications

1. Chen Y, **Groce A**, Zhang C, Wong W-K, Fern X, Eide E, Regehr J. Taming Compiler Fuzzers, ACM SIGPLAN Conference on Programming Language Design and Implementation, 197-208, June 2013.
2. **Groce A**, Kulesza T, Zhang C, Shamasunder S, Burnett M, Wong W K, Stumpf S, Das S, Shinsel A, Bice F, McIntosh F. You are the Only Possible Oracle: Effective Test Selection for End Users of Interactive Machine Learning Systems. IEEE Transactions on Software Engineering, 40(3):307-323, March 2014.
3. Zhang, C, **Groce A**, Alipour M A. Using Test Case Reduction and Prioritization to Improve Symbolic Execution, ACM International Symposium on Software Testing and Analysis, 60-70, July 2014.
4. Gopinath R, Jensen C, **Groce A**. Code Coverage for Suite Evaluation by Developers, ACM/IEEE International Conference on Software Engineering, 72-82, May 2014.
5. Chaki S, Clarke E, **Groce A**, Jha S, Veith H. Modular Verification of Software Components in C. IEEE Transactions on Software Engineering, 30(6): 388-402, June 2004.

### (d) Synergistic Activities

1. Served as member of the program committee for Fundamental Approaches to Software Engineering 2011, NASA Formal Methods Symposium 2011, IEEE/ACM Conference on Automated Software Engineering, 2013, 2019, ACM International Symposium on Software Testing and Analysis, 2014, 2017, ACM/IEEE International Conference on Software Engineering, 2018 and 2019.
2. Currently serving as associate editor for Software Testing, IEEE Software magazine.
3. Contributed to design, development, and maintenance of numerous software testing and analysis tools, including the TSTL Template Scripting Testing Language for Python, the award-winning Java PathFinder from NASA Ames (now an open-source project at Sourceforge), CBMC (from Carnegie Mellon University, ETH Zurich, and Oxford), the MAGIC model checker (CMU), the Concurrency Workbench NC, tools used internally by NASA/JPL, DeepState, Echidna, and Slither. I am the most active developer for TSTL, the universalmutator, and DeepState; projects for which I am lead have over 500 total stars on GitHub, and are used internationally. I make over 1000 GitHub commits and open hundreds of issues per year, in addition to reviewing code for a large number of PRs.
4. Advising IBM Linux kernel developers, NASA software engineers, author of SQLite, maintainers of pyfakefs, developers/test engineers for OpenSSL, and developers of Echidna smart contract fuzzer and Slither smart contract static analysis tool on use of mutation testing to enhance test suites and improve software reliability.
5. Helped design and implement mutation tools for languages beyond Java and C, including the first generally useable tools for Haskell, Python, Swift, Rust, and Solidity mutation testing.