# 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 2020- Associate Editor, ACM Transactions on Software Engineering and Methodology

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

Aug 2019

Aug 2018- Associate Editor for Software Testing, IEEE Software

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. Holmes J, Ahmed I, Brindescu C, Gopinath R, Zhang H, **Groce A**. Using Relative Lines of Code to Guide Automated Test Generation for Python. *ACM Transactions on Software Engineering and Methodology*, 29(4), 28:1-28:38 (2020).
2. **Groce** A, Havelund K, Holzmann G, Joshi R, Xu R. Establishing Flight Software Reliability: Testing, Model-Checking, Constraint-Solving, Monitoring and Learning. *Annals of Mathematics and Artificial Intelligence*, 70(4): 315-348, April 2014.
3. Goodman P, **Groce A**.DeepState: Symbolic Unit Testing for C and C++. NDSS Workshop on Binary Analysis Research, San Diego, CA, February 2018.
4. **Groce A**, Ahmed I, Jensen C, McKenney P, Holmes J. How Verified (or Tested) is My Code? Falsification-Driven Verification and Testing. *Automated Software Engineering Journal*. 25(4): 917-960, December 2018.
5. **Groce A**, Flikkema P, Holmes J. Towards Automated Composition of Heterogenous Test for Cyber-Physical Systems. *Workshop on Testing Embedded and Cyber-Physical Systems*, pages 12-15, Santa Barbara, CA, July 2017.

### 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, 2021, ACM International Symposium on Software Testing and Analysis, 2014, 2017, ACM/IEEE International Conference on Software Engineering, 2018, 2019, 2022.
2. Submitted hundreds of bugs to open source projects, as a result of automated testing efforts; of these the vast majority have been deemed important and fixed, in some cases with bug bounty awarded.
3. Contributed to design and development of numerous software testing and analysis tools, including the DeepState parameterized unit fuzzing system, 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, and the smart contract analysis tools Echidna and Slither
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.