# 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

2015-Present 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, **Groce A**, Pinto J, Mittal P, Azimi P, Kellar K, O’Brien J. TSTL: the Template Scripting Testing Language. International Journal on Software Tools for Technology Transfer, accepted for publication.
2. Alipour M A, Shi A, Gopinath R, Marinov D, **Groce A**. Evaluating non-adequate test-case reduction. IEEE/ACM International Conference on Automated Software Engineering, 16-26, September 2016.
3. **Groce A**, Alipour M A, Zhang C, Chen Y, Regehr J. Cause Reduction: Delta Debugging, Even Without Bugs. Software Testing, Verification, and Reliability, 26(1): 40-68, January 2016.
4. 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.
5. 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.

### Other Significant Publications

1. Holmes J, **Groce A**, Alipour M A. Mitigating (and Exploiting) Test Reduction Slippage. Workshop on Automated Software Testing, November 2016.
2. **Groce A**, Ahmed I, Jensen C, McKenney P. How Verified is my Code? Falsification-Driven Verification. IEEE/ACM International Conference on Automated Software Engineering, 737-748, November 2015.
3. **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.
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, ACM International Symposium on Software Testing, 2014, ACM International Symposium on Software Testing, 2017, ACM/IEEE International Conference on Software Engineering, 2018.
2. Served as co-chair of SPIN Workshop on Model Checking Software, 2011, Workshop on Dynamic Analysis, 2013, Workshop on Causal Reasoning for Embedded and safety-critical Systems Technologies.
3. Contributed to design and development of numerous software testing and analysis tools, including 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.
4. Advising IBM Linux kernel developers, NASA software engineers, author of SQLite, and developers/test engineers for OpenSSL on use of mutation testing to enhance test suites and improve software reliability.
5. Lead for design and implementation of the open source TSTL Template Scripting Testing Language tool for testing Python code, and lead for efforts to use this system to test critical Python libraries and infrastructure, and help developers to build their own TSTL test harnesses for future testing of their systems, as well as educational integration of TSTL into classes on software engineering, including requirements and testing.