

References

- [1] Florent Kirchner, Nikolai Kosmatov, Virgile Prevosto, Julien Signoles, and Boris Yakobowski. Frama-C: A software analysis perspective. *Formal Asp. Comput.*, 27(3):573–609, 2015.
- [2] Michal Zalewski. american fuzzy lop (2.35b). <http://lcamtuf.coredump.cx/afl/>, November 2014.
- [3] Peter Goodman and Alex Groce. DeepState: Symbolic unit testing for C and C++. In *NDSS Workshop on Binary Analysis Research*, 2018.
- [4] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor Comput Sci*, 126(2):183 – 235, 1994.
- [5] Patricia Bouyer, François Laroussinie, Nicolas Markey, Joël Ouaknine, and James Worrell. Timed temporal logics. In *Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday*, volume 10460 of *LNCS*, pages 211–230. Springer, 2017.
- [6] Alastair Reid, Luke Church, Shaked Flur, Sarah de Haas, Maritza Johnson, and Ben Laurie. Towards making formal methods normal: meeting developers where they are, 2020.
- [7] Johan Bengtsson, Kim Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi. Uppaal—a tool suite for automatic verification of real-time systems. In *International Hybrid Systems Workshop*, pages 232–243. Springer, 1995.
- [8] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.
- [9] Patrick Baudin, Jean C. Filliâtre, Thierry Hubert, Claude Marché, Benjamin Monate, Yannick Moy, and Virgile Prevosto. *ACSL: ANSI/ISO C Specification Language*, February 2011. <http://frama-c.cea.fr/acsl.html>.
- [10] Gary T. Leavens, Albert L. Baker, and Clyde Ruby. JML: a Java Modeling Language. In *Formal Underpinnings of Java Workshop (at OOPSLA '98)*, October 1998. <http://www-dse.doc.ic.ac.uk/~sue/oopsla/cfp.html>.
- [11] Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proc. of the 4th ACM Symposium on Principles of Programming Languages (POPL 1977)*, pages 238–252. ACM, 1977.
- [12] Allan Blanchard, Frédéric Loulergue, and Nikolai Kosmatov. Towards Full Proof Automation in Frama-C using Auto-Active Verification. In *Nasa Formal Methods*, *LNCS*, pages 88–105. Springer, 2019.
- [13] Lori A. Clarke and David S. Rosenblum. A historical perspective on runtime assertion checking in software development. *SIGSOFT Softw. Eng. Notes*, 31(3):25–37, May 2006.
- [14] N. Williams, B. Marre, P. Mouy, and M. Roger. PathCrawler: automatic generation of path tests by combining static and dynamic analysis. In *EDCC*, 2005.
- [15] Patrice Godefroid, Nils Klarlund, and Koushik Sen. DART: directed automated random testing. In *Programming Language Design and Implementation*, pages 213–223, 2005.
- [16] RTCA Special Committee 167. Software considerations in airborne systems and equipment certification. Technical Report DO-178-B, RTCA, Inc., 1992.
- [17] Peter Goodman, Gustavo Greico, and Alex Groce. Tutorial: DeepState: Bringing vulnerability detection tools into the development cycle. In *IEEE Cybersecurity Development Conference (SECDEV)*, 2018.
- [18] Eric Gamma and Kent Beck. JUnit 5. <http://junit.org/junit5/>.
- [19] Google Test. <https://github.com/google/googletest>, 2008.
- [20] Nikolai Tillmann and Wolfram Schulte. Parameterized unit tests. In *ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 253–262, 2005.
- [21] Yan Shoshitaishvili, Ruoyu Wang, Christopher Salls, Nick Stephens, Mario Polino, Audrey Dutcher, John Grosen, Siji Feng, Christophe Hauser, Christopher Kruegel, and Giovanni Vigna. Sok: (state of) the art of war: Offensive techniques in binary analysis. In *IEEE Symposium on Security and Privacy*, 2016.

- [22] Nick Stephens, John Grosen, Christopher Salls, Audrey Dutcher, Ruoyu Wang, Jacopo Corbetta, Yan Shoshitaishvili, Christopher Kruegel, and Giovanni Vigna. Driller: Augmenting fuzzing through selective symbolic execution. In *Network and Distributed System Security Symposium*, 2016.
- [23] Yan Shoshitaishvili, Ruoyu Wang, Christophe Hauser, Christopher Kruegel, and Giovanni Vigna. Fomalice - automatic detection of authentication bypass vulnerabilities in binary firmware. In *NDSS*, 2015.
- [24] Mark Mossberg, Felipe Manzano, Eric Hennenfent, Alex Groce, Gustavo Grieco, Josselin Feist, Trent Brunson, and Artem Dinaburg. Manticore: A user-friendly symbolic execution framework for binaries and smart contracts. In *34th IEEE/ACM International Conference on Automated Software Engineering, ASE 2019, San Diego, CA, USA, November 11-15, 2019*, pages 1186–1189. IEEE, 2019.
- [25] Kostya Serebryany. Continuous fuzzing with libfuzzer and addresssanitizer. In *Cybersecurity, Development (SecDev), IEEE*, pages 157–157. IEEE, 2016.
- [26] Jaeseung Choi, Joonun Jang, Choongwoo Han, and Sang Kil Cha. Grey-box concolic testing on binary code. In *International Conference on Software Engineering*, pages 736–747, 2019.
- [27] Peng Chen and Hao Chen. Angora: Efficient fuzzing by principled search. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 711–725, 2018.
- [28] honggfuzz. <https://github.com/google/honggfuzz>, 2010.
- [29] Koen Claessen and John Hughes. QuickCheck: a lightweight tool for random testing of Haskell programs. In *International Conference on Functional Programming*, pages 268–279, 2000.
- [30] Rickard Nilsson, Shane Auckland, Mark Sumner, and Sanjiv Sahayam. ScalaCheck user guide. <https://github.com/rickynils/scalacheck/blob/master/doc/UserGuide.md>, September 2016.
- [31] David R. MacIver. Hypothesis: Test faster, fix more. <http://hypothesis.works/>, March 2013.
- [32] Alex Groce and Jervis Pinto. A little language for testing. In *NASA Formal Methods Symposium*, pages 204–218, 2015.
- [33] Josie Holmes, Alex Groce, Jervis Pinto, Pranjal Mittal, Pooria Azimi, Kevin Kellar, and James O’Brien. TSTL: the template scripting testing language. *International Journal on Software Tools for Technology Transfer*, 20(1):57–78, February 2018.
- [34] Nikolai Tillmann and Jonathan De Halleux. Pex—white box test generation for .NET. In *Tests and Proofs*, pages 134–153, 2008.
- [35] Nikolai Tillmann and Wolfram Schulte. Parameterized unit tests with Unit Meister. In *Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 241–244, 2005.
- [36] Cristian Cadar, Daniel Dunbar, and Dawson Engler. KLEE: unassisted and automatic generation of high-coverage tests for complex systems programs. In *OSDI*, 2008.
- [37] Jack Sun, Daniel Fryer, Ashvin Goel, and Angela Demke Brown. Using declarative invariants for protecting file-system integrity. In *Proceedings of the 6th Workshop on Programming Languages and Operating Systems*, page 6. ACM, 2011.
- [38] Alex Groce. Test harness for testfs. <https://github.com/agroce/testfs>, 2018.
- [39] Robert Bocchino, Timothy Canham, Garth Watney, Leonard Reder, and Jeffrey Levison. F prime: An open-source framework for small-scale flight software systems. In *Small Satellite Conference*, 2018.
- [40] NASA. F prime: A flight-proven, multi-platform, open-source flight software framework. <https://github.com/nasa/fprime>, 2018.
- [41] Dokyung Song, Felicitas Hetzelt, Dipanjan Das, Chad Spensky, Yeoul Na, Stijn Volckaert, Giovanni Vigna, Christopher Kruegel, Jean-Pierre Seifert, and Michael Franz. Periscope: An effective probing and fuzzing framework for the hardware-OS boundary. In *NDSS*, 2019.
- [42] Lionel Blatter, Nikolai Kosmatov, Pascale Le Gall, Virgile Prevosto, and Guillaume Petiot. Static and dynamic verification of relational properties on self-composed C code, 2018.
- [43] Virgile Robles, Nikolai Kosmatov, Virgile Prevosto, Louis Rilling, and Pascale Le Gall. MetAcsl:

- Specification and verification of high-level properties. In Tomáš Vojnar and Lijun Zhang, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 358–364, Cham, 2019. Springer International Publishing.
- [44] Abraham A Clements, Eric Gustafson, Tobias Scharnowski, Paul Grosen, David Fritz, Christopher Kruegel, Giovanni Vigna, Saurabh Bagchi, and Mathias Payer. {HALucinator}: Firmware re-hosting through abstraction layer emulation. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1201–1218, 2020.
 - [45] Alex Groce, Gerard Holzmann, Rajeev Joshi, and Ru-Gang Xu. Putting flight software through the paces with testing, model checking, and constraint-solving. In *Workshop on Constraints in Formal Verification*, pages 1–15, 2008.
 - [46] Alex Groce, Klaus Havelund, Gerard Holzmann, Rajeev Joshi, and Ru-Gang Xu. Establishing flight software reliability: Testing, model checking, constraint-solving, monitoring and learning. *Annals of Mathematics and Artificial Intelligence*, 70(4):315–349, 2014.
 - [47] Marcel Böhme, Van-Thuan Pham, and Abhik Roychoudhury. Coverage-based greybox fuzzing as Markov chain. *IEEE Transactions on Software Engineering*, 45(5):489–506, 2017.
 - [48] Trail of Bits. Fuzzing tips. https://github.com/crytic/building-secure-contracts/blob/master/program-analysis/echidna/fuzzing_tips.md.
 - [49] Caroline Lemieux and Koushik Sen. FairFuzz: a targeted mutation strategy for increasing greybox fuzz testing coverage. In *International Conference on Automated Software Engineering*, pages 475–485, 2018.
 - [50] Sanjay Rawat, Vivek Jain, Ashish Kumar, Lucian Cojocar, Cristiano Giuffrida, and Herbert Bos. VUzzer: application-aware evolutionary fuzzing. In *NDSS (Network and Distributed Security Symposium)*, 2017.
 - [51] Lei Zhao, Yue Duan, Heng Yin, and Jifeng Xuan. Send hardest problems my way: Probabilistic path prioritization for hybrid fuzzing. In *NDSS (Network and Distributed Security Symposium)*, 2019.
 - [52] Cornelius Aschermann, Sergej Schumilo, Tim Blazytko, Robert Gawlik, and Thorsten Holz. REDQUEEN: fuzzing with input-to-state correspondence. In *NDSS (Network and Distributed Security Symposium)*, 2019.
 - [53] Daniel Kroening, Edmund M. Clarke, and Flavio Lerda. A tool for checking ANSI-C programs. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 168–176, 2004.
 - [54] Andreas Tiemeyer, Tom Melham, Daniel Kroening, and John O’Leary. Crest: Hardware formal verification with ansi-c reference specifications, 2019.
 - [55] Byron Cook, Kareem Khazem, Daniel Kroening, Serdar Tasiran, Michael Tautschnig, and Mark R Tuttle. Model checking boot code from aws data centers. *Formal Methods in System Design*, pages 1–19, 2020.
 - [56] Alex Groce, Iftekhhar Ahmed, Carlos Jensen, Paul E McKenney, and Josie Holmes. How verified (or tested) is my code? falsification-driven verification and testing. *Automated Software Engineering Journal*, 25(4):917–960, 2018.
 - [57] Gerard J. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley Professional, 2003.
 - [58] Gerard Holzmann and Rajeev Joshi. Model-driven software verification. In *SPIN Workshop on Model Checking of Software*, pages 76–91, 2004.
 - [59] Gerard Holzmann, Rajeev Joshi, and Alex Groce. Model driven code checking. *Automated Software Engineering*, 15(3–4):283–297, 2008.
 - [60] Alex Groce and Rajeev Joshi. Random testing and model checking: Building a common framework for nondeterministic exploration. In *Workshop on Dynamic Analysis*, pages 22–28, 2008.
 - [61] Alex Groce and Martin Erwig. Finding common ground: Choose, assert, and assume. In *International Workshop on Dynamic Analysis*, pages 12–17, 2012.
 - [62] Giovanni Liva, Muhammad Taimoor Khan, and Martin Pinzger. Extracting timed automata from java methods. In *2017 IEEE 17th International Working Conference on Source Code Analysis and*

- Manipulation (SCAM)*, pages 91–100. IEEE, 2017.
- [63] Truong X. Nghiem. MLE+: A Matlab-EnergyPlus Co-simulation Interface. <https://github.com/nxtruong/mle-legacy>, July 2022.
 - [64] Willy Bernal, Madhur Behl, Truong X. Nghiem, and Rahul Mangharam. MLE+: A tool for integrated design and deployment of energy efficient building controls. In *Proceedings of the 4th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys'12)*, pages 123–130, Toronto, Ontario, Canada, November 2012. ACM.
 - [65] Truong X. Nghiem. OpenBuildNet Co-Simulation Platform. <https://github.com/nxt-lab/openBuildNet>, July 2022.
 - [66] Truong X. Nghiem, Altüg Bitlisliöglu, Tomasz Gorecki, Faran A. Qureshi, and Colin N. Jones. Open-BuildNet Framework for Distributed Co-Simulation of Smart Energy Systems. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6, November 2016.
 - [67] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–386, June 2005.
 - [68] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12):399, 2013.
 - [69] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 92–106. Springer, 2010.
 - [70] Yashwanth Annpureddy, Che Liu, Georgios Fainekos, and Sriram Sankaranarayanan. S-taliro: A tool for temporal logic falsification for hybrid systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 254–257. Springer, 2011.
 - [71] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.
 - [72] George Klees, Andrew Ruef, Benji Cooper, Shiyi Wei, and Michael Hicks. Evaluating fuzz testing. *arXiv preprint arXiv:1808.09700*, 2018.
 - [73] James R. Wilcox, Doug Woos, Pavel Panchekha, Zachary Tatlock, Xi Wang, Michael D. Ernst, and Thomas Anderson. Verdi: A framework for implementing and formally verifying distributed systems. In *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 357–368, New York, NY, USA, 2015. ACM.
 - [74] Doug Woos, James R. Wilcox, Steve Anton, Zachary Tatlock, Michael D. Ernst, and Thomas Anderson. Planning for change in a formal verification of the raft consensus protocol. In *Proceedings of the 5th ACM SIGPLAN Conference on Certified Programs and Proofs (CPP)*, pages 154–165, New York, NY, USA, 2016. ACM.
 - [75] Christine Paulin-Mohring. Modelisation of timed automata in coq. In *Theoretical Aspects of Computer Software, 4th International Symposium, TACS 2001, Sendai, Japan, October 29-31, 2001, Proceedings*, volume 2215 of *LNCS*, pages 298–315. Springer, 2001.
 - [76] Anders Hessel, Kim Larsen, Marius Mikucionis, Brian Nielsen, Paul Pettersson, and Arne Skou. Testing Real-Time systems using UPPAAL. In *Formal Methods and Testing*, pages 77–117. 2008.
 - [77] Anders Hessel and Paul Pettersson. CoVer - a real-time test case generation tool. In *19th IFIP International Conference on Testing of Communicating Systems and 7th International Workshop on Formal Approaches to Testing of Software*, 2007.
 - [78] Kim G. Larsen, Marius Mikucionis, and Brian Nielsen. Testing real-time embedded software using UPPAAL-TRON: an industrial case study. In *the 5th ACM international conference on Embedded software*, pages 299 – 306. ACM Press New York, NY, USA, September 18–22 2005.
 - [79] Mark Guzdial. A media computation course for non-majors. In *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE '03*, pages 104–108,

New York, NY, USA, 2003. ACM.

- [80] Alex Groce, Rijnard van Tonder, Goutamkumar Tulajappa Kalburgi, and Claire Le Goues. Making no-fuss compiler fuzzing effective. In *ACM SIGPLAN International Conference on Compiler Construction*, 2022. accepted for publication.
- [81] Alex Groce, Kush Jain, , Rijnard van Tonder, Goutamkumar Tulajappa Kalburgi, and Claire Le Goues. Looking for lacunae in bitcoin core’s fuzzing efforts. In *ACM/IEEE International Conference on Software Engineering*, 2022. accepted for publication.
- [82] Alex Groce, Goutamkumar Tulajappa Kalburgi, Claire Le Goues, Kush Jain, and Rahul Gopinath. First, fuzz the mutants. In *1st International Fuzzing Workshop*, 2022. accepted for publication.