References

1. W. H. K. Bester, C. P. Inggs, and W. C. Visser, *Test-case generation and bug-finding through symbolic execution*, Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (New York, NY, USA), SAICSIT '12, ACM, 2012, pp. 1–9.
2. Cristian Cadar, Patrice Godefroid, Sarfraz Khurshid, Corina S. Păsăreanu, Koushik Sen, Nikolai Tillmann, and Willem Visser, *Symbolic execution for software testing in practice: Preliminary assessment*, Proceedings of the 33rd International Conference on Software Engineering (New York, NY, USA), ICSE '11, ACM, 2011, pp. 1066–1071.
3. Satish Chandra, Colin S. Gordon, Jean-Baptiste Jeannin, Cole Schlesinger, Manu Sridharan, Frank Tip, and Youngil Choi, *Type inference for static compilation of javascript*, SIGPLAN Not. **51** (2016), no. 10, 410–429.
4. Hanfeng Chen, Alexander Krolik, Erick Lavoie, and Laurie J. Hendren, *Automatic vectorization for MATLAB*, LCPC, Lecture Notes in Computer Science, vol. 10136, Springer, 2016, pp. 171–187.
5. Christoph Csallner, Yannis Smaragdakis, and Tao Xie, *Dsd-crasher: A hybrid analysis tool for bug finding*, ACM Trans. Softw. Eng. Methodol. **17** (2008), no. 2, 8:1–8:37.
6. Anton Willy Dubrau and Laurie Jane Hendren, *Taming matlab*, Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications (New York, NY, USA), OOPSLA '12, ACM, 2012, pp. 503–522.
7. Vincent Foley-Bourgon and Laurie Hendren, *Efficiently implementing the copy semantics of matlab's arrays in javascript*, Proceedings of the 12th Symposium on Dynamic Languages (New York, NY, USA), DLS 2016, ACM, 2016, pp. 72–83.
8. Patrice Godefroid, Michael Y. Levin, and David Molnar, *Sage: Whitebox fuzzing for security testing*, Commun. ACM **55** (2012), no. 3, 40–44.
9. Robert Guo, *Mongodb's javascript fuzzer*, Commun. ACM **60** (2017), no. 5, 43–47.
10. Philipp Haller and Alex Loiko, *Lacasa: Lightweight affinity and object capabilities in scala*, Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications (New York, NY, USA), OOPSLA 2016, ACM, 2016, pp. 272–291.
11. Prateek Saxena, Devdatta Akhawe, Steve Hanna, Feng Mao, Stephen McCamant, and Dawn Song, *A symbolic execution framework for javascript*, Proceedings of the 2010 IEEE Symposium on Security and Privacy (Washington, DC, USA), SP '10, IEEE Computer Society, 2010, pp. 513–528.
12. Koushik Sen, George Necula, Liang Gong, and Wontae Choi, *Multise: Multi-path symbolic execution using value summaries*, Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (New York, NY, USA), ESEC/FSE 2015, ACM, 2015, pp. 842–853.