Advances in Automated Program Repair and a Call to Arms

Westley Weimer

University of Virginia
weimer@virginia.edu

Abstract. In this keynote address I survey recent success and momentum in the subfield of automated program repair. I also encourage the search-based software engineering community to rise to various challenges and opportunities associated with test oracle generation, large-scale human studies, and reproducible research through benchmarks.

I discuss recent advances in *automated program repair*, focusing on the search-based GenProg technique but also presenting a broad overview of the subfield. I argue that while many automated repair techniques are "correct by construction" or otherwise produce only a single repair (e.g., AFix [13], Axis [17], Coker and Hafiz [4], Demsky and Rinard [7], Gopinath *et al.* [12], Jolt [2], Juzi [8], etc.), the majority can be categorized as "generate and validate" approaches that enumerate and test elements of a space of candidate repairs and are thus directly amenable to search-based software engineering and mutation testing insights (e.g., ARC [1], AutoFix-E [23], ARMOR [3], CASC [24], ClearView [21], Debroy and Wong [6], FINCH [20], PACHIKA [5], PAR [14], SemFix [18], Sidiroglou and Keromytis [22], etc.). I discuss challenges and advances such as scalability, test suite quality, and repair quality while attempting to convey the excitement surrounding a subfield that has grown so quickly in the last few years that it merited its own session at the 2013 International Conference on Software Engineering [3,4,14,18]. Time permitting, I provide a frank discussion of mistakes made and lessons learned with GenProg [15].

In the second part of the talk, I pose three challenges to the SBSE community. I argue for the importance of *human studies* in automated software engineering. I present and describe multiple "how to" examples of using crowdsourcing (e.g., Amazon's Mechanical Turk) and massive online education (MOOCs) to enable SBSE-related human studies [10,11]. I argue that we should leverage our great strength in testing to tackle the increasingly-critical problem of test *oracle generation* (e.g., [9]) — not just test data generation — and draw supportive analogies with the subfields of specification mining and invariant detection [16,19]. Finally, I challenge the SBSE community to facilitate reproducible research and scientific advancement through *benchmark* creation, and support the need for such efforts with statistics from previous accepted papers.

G. Ruhe and Y. Zhang (Eds.): SSBSE 2013, LNCS 8084, pp. 1–3, 2013.

References

1. Bradbury, J.S., Jalbert, K.: Automatic repair of concurrency bugs. In: International Symposium on Search Based Software Engineering - Fast Abstracts, pp. 1–2 (September 2010)
2. Carbin, M., Misailovic, S., Kling, M., Rinard, M.C.: Detecting and escaping infinite loops with jolt. In: Mezini, M. (ed.) ECOOP 2011. LNCS, vol. 6813, pp. 609–633. Springer, Heidelberg (2011)
3. Carzaniga, A., Gorla, A., Mattavelli, A., Perino, N., Pezzè, M.: Automatic recovery from runtime failures. In: International Conference on Sofware Engineering (2013)
4. Coker, Z., Hafiz, M.: Program transformations to fix C integers. In: International Conference on Sofware Engineering (2013)
5. Dallmeier, V., Zeller, A., Meyer, B.: Generating fixes from object behavior anomalies. In: Automated Software Engineering, pp. 550–554 (2009)
6. Debroy, V., Wong, W.E.: Using mutation to automatically suggest fixes for faulty programs. In: International Conference on Software Testing, Verification, and Validation, pp. 65–74 (2010)
7. Demsky, B., Ernst, M.D., Guo, P.J., McCamant, S., Perkins, J.H., Rinard, M.C.: Inference and enforcement of data structure consistency specifications. In: International Symposium on Software Testing and Analysis (2006)
8. Elkarablieh, B., Khurshid, S.: Juzi: A tool for repairing complex data structures. In: International Conference on Software Engineering, pp. 855–858 (2008)
9. Fraser, G., Zeller, A.: Mutation-driven generation of unit tests and oracles. Transactions on Software Engineering 38(2), 278–292 (2012)
10. Fry, Z.P., Landau, B., Weimer, W.: A human study of patch maintainability. In: Heimdahl, M.P.E., Su, Z. (eds.) International Symposium on Software Testing and Analysis, pp. 177–187 (2012)
11. Fry, Z.P., Weimer, W.: A human study of fault localization accuracy. In: International Conference on Software Maintenance, pp. 1–10 (2010)
12. Gopinath, D., Malik, M.Z., Khurshid, S.: Specification-based program repair using SAT. In: Abdulla, P.A., Leino, K.R.M. (eds.) TACAS 2011. LNCS, vol. 6605, pp. 173–188. Springer, Heidelberg (2011)
13. Jin, G., Song, L., Zhang, W., Lu, S., Liblit, B.: Automated atomicity-violation fixing. In: Programming Language Design and Implementation (2011)
14. Kim, D., Nam, J., Song, J., Kim, S.: Automatic patch generation learned from human-written patches. In: International Conference on Sofware Engineering (2013)
15. Le Goues, C., Dewey-Vogt, M., Forrest, S., Weimer, W.: A systematic study of automated program repair: Fixing 55 out of 105 bugs for $8 each. In: International Conference on Software Engineering, pp. 3–13 (2012)
16. Le Goues, C., Weimer, W.: Measuring code quality to improve specification mining. IEEE Transactions on Software Engineering 38(1), 175–190 (2012)
17. Liu, P., Zhang, C.: Axis: Automatically fixing atomicity violations through solving control constraints. In: International Conference on Software Engineering, pp. 299–309 (2012)
18. Nguyen, H.D.T., Qi, D., Roychoudhury, A., Chandra, S.: SemFix: Program repair via semantic analysis. In: International Conference on Sofware Engineering, pp. 772–781 (2013)
19. Nguyen, T., Kapur, D., Weimer, W., Forrest, S.: Using dynamic analysis to discover polynomial and array invariants. In: International Conference on Software Engineering, pp. 683–693 (2012)

20. Orlov, M., Sipper, M.: Flight of the FINCH through the Java wilderness. Transactions on Evolutionary Computation 15(2), 166–192 (2011)
21. Perkins, J.H., Kim, S., Larsen, S., Amarasinghe, S., Bachrach, J., Carbin, M., Pacheco, C., Sherwood, F., Sidiroglou, S., Sullivan, G., Wong, W.-F., Zibin, Y., Ernst, M.D., Rinard, M.: Automatically patching errors in deployed software. In: Symposium on Operating Systems Principles (2009)
22. Sidiroglou, S., Keromytis, A.D.: Countering network worms through automatic patch generation. IEEE Security and Privacy 3(6), 41–49 (2005)
23. Wei, Y., Pei, Y., Furia, C.A., Silva, L.S., Buchholz, S., Meyer, B., Zeller, A.: Automated fixing of programs with contracts. In: International Symposium on Software Testing and Analysis, pp. 61–72 (2010)
24. Wilkerson, J.L., Tauritz, D.R., Bridges, J.M.: Multi-objective coevolutionary automated software correction. In: Genetic and Evolutionary Computation Conference, pp. 1229–1236 (2012)