A Correlation Study between Automated Program Repair and Test-Suite Metrics

Jooyong Yi
Innopolis University
j.yi@innopolis.ru

Shin Hwei Tan
National University of Singapore
shinhwei@comp.nus.edu.sg

Sergey Mechtaev
National University of Singapore
mechtaev@comp.nus.edu.sg

Marcel Böhme
National University of Singapore
mboehme@comp.nus.edu.sg

Abhik Roychoudhury
National University of Singapore
abhik@comp.nus.edu.sg

ABSTRACT

Automated program repair has attracted attention due to its potential to reduce debugging cost. Prior works show the feasibility of automated repair, and the research focus is gradually shifting towards the quality of generated patches. One promising direction is to control the quality of generated patches by controlling the quality of test-suites used. In this paper, [1]we investigate the question: "Can traditional test-suite metrics used in software testing be used for automated program repair?". We empirically investigate the effectiveness of test-suite metrics (statement / branch coverage and mutation score) in controlling the reliability of repairs (the likelihood that repairs cause regressions). We conduct the largest-scale experiments to date with real-world software, and perform the first correlation study between test-suite metrics and the reliability of generated repairs. Our results show that by increasing test-suite metrics, the reliability of repairs tend to increase. Particularly, such trend is most strongly observed in statement coverage. This implies that traditional test-suite metrics used in software testing can also be used to improve the reliability of repairs in program repair.

ACM Reference Format:
Jooyong Yi, Shin Hwei Tan, Sergey Mechtaev, Marcel Böhme, and Abhik Roychoudhury. 2018. A Correlation Study between Automated Program Repair and Test-Suite Metrics. In *ICSE '18: ICSE '18: 40th International Conference on Software Engineering , May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3180155.3182517

1 OVERVIEW

Automated program repair approaches have demonstrated recent success in fixing real-world software [3–5, 7, 10]. Instead of the feasibility of repair techniques, recent studies focus on the *correctness* of patches — patches that pass all provided tests and also indeed fixes the bug [8, 9]. Most repair approaches use test-suites as proxies for software specification. As test-suites are incomplete specifications, generated repairs may be incomplete. Despite this limitation, software quality could be improved by enhancing the quality of given test suite. This motivates our key research question — *is it possible to control the quality of automatically generated repair by improving the quality of test-suite?* Moreover, we also study how test-suite metrics affect repairability and repair time.

We conduct large-scale experiments on the correlation between test-suite quality and automated repair by evaluating four large real-world programs and SIR benchmark [1]. Compared to prior study that were evaluated on small programs [6], our study provide stronger empirical evidences on the correlation between the test-suites quality and the quality of generated repairs. For the fist time, we also compare various test-suite metrics (statement coverage, branch coverage, test-suite size, and mutation score), focusing on their degrees of correlation (i.e., correlation coefficients) with repair quality. Our study investigates whether traditional test-suite metrics used in software testing are also useful in the context of automated repair, and which test-suite metric is the most effective. We measure the quality of repairs by computing *reliability* (whether generated repairs cause test failures in the held-out test suite). We obtained repairs generated from GenProg [2, 10] and SemFix [5].

Our results show that traditional test-suite metrics are *negatively correlated* with the likelihood that a repair causes regressions (*regression ratio*). This implies that the traditional test-suite metrics proposed for software testing can also be used for automated program repair. Among the evaluated test-suite metrics, statement coverage is the most strongly correlated with regression ratio.

[1]The original journal paper is published as EMSE-D-16-00209R2 [11]

ICSE '18, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5638-1/18/05.
https://doi.org/10.1145/3180155.3182517

REFERENCES

[1] H. Do, S. Elbaum, and G. Rothermel. 2005. Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact. *EMSE* (2005), 405–435.
[2] C. Le Goues, T. Nguyen, S. Forrest, and W. Weimer. 2012. GenProg: A generic method for automatic software repair. *TSE* 38, 1 (2012), 54–72.
[3] S. Mechtaev, J. Yi, and A. Roychoudhury. 2015. DirectFix: Looking for Simple Program Repairs *(ICSE '15)*, Vol. 1. 448–458.
[4] S. Mechtaev, J. Yi, and A. Roychoudhury. 2016. Angelix: Scalable Multiline Program Patch Synthesis via Symbolic Analysis *(ICSE '16)*. ACM, 691–701.
[5] H. D. T. Nguyen, D. Qi, A. Roychoudhury, and S. Chandra. 2013. SemFix: Program repair via semantic analysis *(ICSE'13)*. IEEE, 772–781.
[6] E. K Smith, E. T Barr, C. Le Goues, and Y. Brun. 2015. Is the cure worse than the disease? overfitting in automated program repair. In *FSE'15*. 532–543.
[7] Shin Hwei Tan and Abhik Roychoudhury. 2015. Relifix: Automated Repair of Software Regressions *(ICSE '15)*. ACM, 471–482.
[8] S. H. Tan, Hiroaki Y., M. R Prasad, and A. Roychoudhury. 2016. Anti-patterns in search-based program repair. In *FSE*. ACM, 727–738.
[9] S. H. Tan, J. Yi, Yulis, S. Mechtaev, and A. Roychoudhury. 2017. Codeflaws: a programming competition benchmark for evaluating automated program repair tools. In *ICSE Companion*. 180–182.
[10] W. Weimer, Z.P. Fry, and S. Forrest. 2013. Leveraging program equivalence for adaptive program repair: Models and first results. In *ASE*.
[11] J. Yi, S. H. Tan, S. Mechtaev, M. Böhme, and A. Roychoudhury. 2017. A Correlation Study between Automated Program Repair and Test-Suite Metrics. *EMSE* (2017).