Estimating the Potential of Program Repair Search Spaces with Commit Analysis (journal-first talk)

Khashayar Etemadi
School of EECS
KTH Royal Institute of Technology
Stockholm, Sweden
khaes@kth.se

Niloofar Tarighat
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
tarighat_niloofar@ee.sharif.edu

Siddharth Yadav
Computer Science and Engineering Department
Indraprastha Institute of Information Technology
Delhi, India
siddharth16268@iiitd.ac.in

Matias Martinez
Computer Science Department
Université Polytechnique Hauts-de-France
Lille, France
matias.martinez@uphf.fr

Martin Monperrus
School of EECS
KTH Royal Institute of Technology
Stockholm, Sweden
monperrus@kth.se

I. Extended Abstract

The most natural method for evaluating program repair systems is to run them on bug datasets, such as Defects4J. There are two main obstacles when this evaluation is done on an arbitrary software project. First, fully executing a repair approach on a real world project often requires heavy and time-consuming configuration of the repair approach and the target project. Second, the target programs under repair should have a test suite specifying their correct behavior and, at least, one failing test case that exposes the bug. Previous work showed it is hard to find real world commits with test suites that can be compiled and executed. These two major obstacles (configuration and dependability on strong testing) hinder assessment of automated program repair on new projects.

In this paper, we propose a purely static method to evaluate the potential of the search space of repair approaches. In this context, the search space of the repair approach is the set of all program patches that can be potentially generated by a repair approach. Our new method enables researchers and practitioners to encode the search spaces of repair approaches and select potentially useful ones without struggling with tool configuration and execution. We encode the search spaces by specifying the repair strategies they employ. Next, we use the specifications to check whether past commits lie in repair search spaces. For a repair approach, including many human-written past commits in its search space indicates its potential to generate useful patches. We implement our evaluation method in LighterR. LighterR gets a Git repository and outputs a list of commits whose source code changes lie in repair search spaces.

Our original paper can be found via https://doi.org/10.1016/j.jss.2022.111263.

To sum up, our contributions are:

- A novel method for specifying the search space of program repair approaches, appropriate to study the potential of program repair to create patches corresponding to past commits of software repositories. This framework implemented in a tool called LighterR, is lightweight, it does not require configuration and execution of repair systems.
- A comprehensive series of experiments on past commits. By analyzing $55,309$ human-written commits from 72 Github repositories, we show that 1.35% $(747/55,309)$ of past commits lie in the search space of at least one of the considered repair systems, 62% of these commits are indeed bug-fixing commits according to the manual inspection we conducted. Overall, our experiments show that our novel method is an effective means to study the potential of program repair search space.
- A systematic measurement of the reliability of LighterR. Our prototype system has a precision and recall of 77% and 92%, respectively, which is arguably high compared to close tools, such as PPD (Patch Pattern Detector).

All the data presented in this paper are publicly-available at https://github.com/khaes-kth/GithubRepairPatterns.

II. Journal-First Eligibility

This paper contains 100% novel content and is not an extension of a workshop or conference paper. This paper was accepted on 03 February 2022 for the "Research Paper" track of the Journal of Systems and Software.

III. Full bibliographic reference

Khashayar Etemadi, Niloofar Tarighat, Siddharth Yadav, Matias Martinez, Martin Monperrus, Estimating the potential of program repair search spaces with commit analysis, Journal of Systems and Software, Volume 188, 2022, 111263, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2022.111263.