1. **CREX: Predicting patch correctness in automated repair of C programs through transfer learning of execution semantics**

Yan, Dapeng; Liu, Kui; Niu, Yuqing; Li, Li; Liu, Zhe; Liu, Zhiming; Klein, Jacques; Bissyandé, Tegawendé F. **Source:** *Information and Software Technology*, v 152, December 2022; **ISSN:** 09505849; **DOI:** 10.1016/j.infsof.2022.107043; **Article number:** 107043; **Publisher:** Elsevier B.V.

Author affiliation:
Nanjing University of Aeronautics and Astronautics, Nanjing, China
Huawei Software Engineering Application Technology Lab, Hangzhou, China
Monash University, Melbourne, Australia
Northwestern Polytechnical University, Xian, China
University of Luxembourg, Luxembourg City, Luxembourg

Abstract:

A significant body of automated program repair literature relies on test suites to assess the validity of generated patches. Because such oracles are weak, state-of-the-art repair tools can validate some patches that overfit the test cases but are actually incorrect. This situation has become a prime concern in APR, hindering its adoption by the industry. This work investigates execution semantic features based on micro-traces, a form of under-constrained dynamic traces. We build on transfer learning to explore function code representations that are amenable to semantic similarity computation and can therefore be leveraged for classifying patch correctness. Our CREX prototype implementation is based on the TREX framework. Experimental results on patches generated by the CoCoNut APR tool on CodeFlaws programs indicate that our approach can yield high accuracy in predicting patch correctness. The learned embeddings were proven to capture semantic similarities between functions, which was instrumental in training a classifier that identifies patch correctness by learning to discriminate between correctly patched code and incorrectly patched code based on their semantic similarity with the buggy function.

Main Heading: Semantics **Controlled terms:** C (programming language) - Repair - Software testing

Uncontrolled terms: C programs - Execution semantics - Patch correctness - Program repair - Repair tools - Semantic features - Semantic similarity - State of the art - Test case - Transfer learning

Classification Code: 723.1.1 Computer Programming Languages - 723.5 Computer Applications - 913.5 Maintenance

Funding details: Number: 2020A06, Acronym: -, Sponsor: -; Number: 62172214, Acronym: NSFC, Sponsor: National Natural Science Foundation of China; Number: BK20210279, Acronym: -, Sponsor: Natural Science Foundation of Jiangsu Province; Number: 2020AAA0107704, Acronym: NKRDPC, Sponsor: National Key Research and Development Program of China;

Funding text: This work was supported by the National Natural Science Foundation of China (Grant No. 62172214), the National Key R&D Program of China (No. 2020AAA0107704), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20210279), and the Open Project Program of the State Key Laboratory of Mathematical Engineering and Advanced Computing (No. 2020A06).

Database: Compendex