PatchID: An Overfitting Patches Identification

Method for Automated Program Repair

Xuan Zhou1,2, Xingqi Wang1,2\*†

1\*School of Computer Science, Hangzhou Dianzi University,

Street, Hangzhou, 310018, Zhejiang, China.

2 Key Laboratory of Discrete Industrial Internet of Things of Zhejiang Province, Hangzhou, 310018, Zhejiang, China.

\*Corresponding author(s). E-mail(s): [xqwang@hdu.edu.cn;](mailto:xqwang@hdu.edu.cn;)

Contributing authors: 212050276@hdu.edu.cn

abstract： Automated Program Repair (APR) needs to verify the correctness of patches after they are generated. Typically, APR uses a test suite as the standard for patch correctness verification. However, the test suite is unable to fully represent the oracle of the program, which causes APR to generat a large number of overfitting patches that not only fail to fix the original error, but also cause new errors. To help developers identify overfitting patches, this paper proposes an overfitting patch identification method called PatchID. The core idea of PatchID is that the dynamic behavior of the passing tests between the buggy program and the correct patch is the same, however the dynamic behavior of the failing tests between them is different. The algorithm first constructs the dynamic behavior expressions that cause the bug from the program and the test suite, then generates new tests to enhance the original test suite, and finally gets the same dynamic behavior expressions from the patch, and identifies whether the patch is overfitting according to whether the value of the dynamic behavior expression changes with the use of the patch. The paper is evaluated on two datasets consisting of 155 Defects4J patches and 365 Java + JML patches, respectively. PatchID successfully identifies 93 overfitting patches and 9 correct patches on the first dataset, and 225 overfitting patches on the second dataset. In addition, PatchID classifies overfitting patches into three types of patches. Experiments show that the method proposed in this paper is superior to the existing similar methods.

**Keywords:** automatic program repair, overfitting patch, program state abstract, test generation

**Statements and Declarations:**

This is the first time we have submitted our manuscript in this journal, and we do not have any financial interests and competing Interests.