Mutation Operators for Feature-Oriented Software Product Lines

Jacob Krüger
Otto-von-Guericke University
Magdeburg, Germany
jkrueger@ovgu.de

Mustafa Al-Hajjaji
pure-systems GmbH
Magdeburg, Germany
mustafa.alhajjaji@pure-systems.com

Thomas Leich
Harz University & METOP GmbH
Wernigerode & Magdeburg, Germany
tleich@hs-harz.de

Gunter Saake
Otto-von-Guericke University
Magdeburg, Germany
saake@ovgu.de

ABSTRACT

In this extended abstract, we describe the *Journal First* summary of our article with the same title published in the Journal of Software: Testing, Verification and Reliability (STVR) [1].

CCS CONCEPTS

• **Software and its engineering → Software product lines**; **Software testing and debugging**.

KEYWORDS

Mutation testing, software product line, variability faults

ACM Reference Format:
Jacob Krüger, Mustafa Al-Hajjaji, Thomas Leich, and Gunter Saake. 2019. Mutation Operators for Feature-Oriented Software Product Lines. In *23rd International Systems and Software Product Line Conference - Volume A (SPLC '19), September 9–13, 2019, Paris, France.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3336294.3342372

Software product lines allow to systematically reuse software artifacts to implement a set of similar variants from the same codebase. The software artifacts are assigned to features that describe a specific functionality and can be enabled or disabled to customize each variant. This variability introduces additional complexity, increasing the costs to test a software product line to ensure the correct behavior of derived variants. Consequently, several testing and sampling techniques have been adapted to improve the efficiency and effectiveness of software-product-line testing.

A particularly challenging technique is mutation testing, where faults are automatically injected into a system to assess the quality of the test suite (not the system itself). The test suite is considered to be appropriate if it kills (its test cases fail) the generated mutants. However, mutation testing itself is an expensive technique, as a large number of mutants can be generated, and each must be run against all tests. For software product lines, this problem becomes even more drastic, as the number of possible variants that must be mutated and tested can exponentially increase with each feature.

Researchers have proposed several approaches to tackle this explosion of costs, mainly cost reduction techniques for mutation testing and sampling variants of a software product line. In our article [1], we propose a set of seven mutation operators for feature-oriented programming that we designed to inject variability faults. Precisely, our operators mutate the mapping between feature model and artifacts as well as variability in the source code. Our idea is to reduce the costs of testing features and their interactions by limiting the number of injected mutations and focusing on actual variability faults. As our operators mutate variability in a software product line, we refer to such operators as *variability-aware mutation operators*. We derived most of these operators from existing ones (e.g., for preprocessors) by adopting them for feature-oriented programming.

To evaluate our mutation operators, we conducted an extensive empirical analysis, including four software product lines as subject systems and comparing our variability-aware with conventional (not specifically designed for variability) mutation operators. The results indicate that our proposed operators are indeed injecting variability faults more controlled than conventional operators. However, we also find that our operators can lead to variants that we cannot compile and that the number of redundant and equivalent mutants is problematic. Consequently, improved cost reduction techniques are necessary to address mutation testing of software product lines. Besides this experience, we also learned that the usability of variability-aware operators heavily depends on how developers employ the variability mechanism (i.e., feature-oriented programming), that automation for such operators is needed, and that test cases must consider the whole software product line. Namely, test cases must be variable by themselves to allow automated mutation testing and they have to test variability to kill the injected mutants. We faced problems with these two issues, as the tests of our subject systems did only partly test variability and were sometimes not usable for all variants.

Acknowledgments. This research is supported by the German Research Foundation (LE 3382/2-1, LE 3382/2-3, SA 465/49-1, SA 465/49-3) and Volkswagen Financial Services AG.

REFERENCES

[1] Jacob Krüger, Mustafa Al-Hajjaji, Thomas Leich, and Gunter Saake. 2019. Mutation Operators for Feature-Oriented Software Product Lines. *Software Testing, Verification and Reliability* 29, 1-2 (2019), e1676. https://doi.org/10.1002/stvr.1676

SPLC '19, September 9–13, 2019, Paris, France

ACM ISBN 978-1-4503-7138-4/19/09.
https://doi.org/10.1145/3336294.3342372