FLACK: Counterexample-Guided Fault Localization for Alloy Models

Guolong Zheng[1], ThanhVu Nguyen[5], Simón Gutiérrez Brida[2,4], Germán Regis[2], Marcelo F. Frias[3,4], Nazareno Aguirre[2,4], and Hamid Bagheri[1]

[1] University of Nebraska-Lincoln, USA
[2] Universidad Nacional de Río Cuarto, Argentina
[3] Instituto Tecnológico de Buenos Aires, Argentina
[4] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
[5] George Mason University, USA

Abstract

Alloy is a specification language that has been used in a wide range of applications, such as program verification, test case generation, IoT and Android security, etc. Unlike imperative languages, such as C or Java, Alloy is declarative, which describes the logic of a computation without describing its control flow and does not generate traces during the execution. Thus, traditional fault localization techniques developed for imperative programs based on analyzing the control flows of passing and failing tests do not directly apply to Alloy. To aid developers in debugging Alloy models, we develop FLACK, a tool to automatically localize Alloy buggy expressions.

Given an Alloy model with violated assertions, FLACK automatically outputs a ranking list of expressions based on their spaciousness to the assertions violations. For each assertion, FLACK first queries the Alloy analyzer for counterexamples, i.e. instances of the model that violate the asserted property. FLACK then uses a Partial Max-SAT (PMAXSAT) solver to find instances that satisfy the asserted property and are most similar to the counterexamples. FLACK then identifies the relations and atoms that are different between the counterexamples and the satisfying instances. The differences illustrate how the counterexamples violate the assertion. The PMAXSAT solver guarantees that these differences are "minimal", containing only essential information related to the assertion violation. By finding expressions most related to these differences, FLACK identifies the potential expressions causing the assertion violation.

FLACK is different than the state of the art on Alloy fault localization in that it does not rely on unit tests which are not commonly found accompanying Alloy models. Instead, FLACK relies on assertions and constraint solvers to obtain counterexamples and satisfying instances, which are the main underlying technology in Alloy and commonly used by the Alloy developers.

We evaluated FLACK on 157 Alloy models collected from previous Alloy fault localization work and popular Alloy models of complex systems.The experiment results show that FLACK can consistently rank the buggy expressions in the top 2% of the suspicious list, with a run time under 5 seconds for most of the models.

This work was published at *43rd International Conference on Software Engineering* on May 2021. https://doi.org/10.1109/ICSE43902.2021.00065