Fuzzing Class Specifications

Facundo Molina[1,2], Marcelo d'Amorim[3], and Nazareno Aguirre[1,2]

[1] Universidad Nacional de Río Cuarto, Argentina
[2] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
[3] Universidade Federal de Pernambuco, Brazil

Abstract

Expressing class specifications via executable constraints is important for various software engineering tasks such as test generation, bug finding and automated debugging, but developers rarely write them. Techniques that infer specifications from code exist to fill this gap, but they are designed to support specific kinds of assertions and are difficult to adapt to support different assertion languages, e.g., to add support for quantification, or additional comparison operators, such as membership or containment.

To address the above issue, we propose SpecFuzzer, a novel technique that combines grammar-based fuzzing, dynamic invariant detection, and mutation analysis, to automatically produce class specifications. SpecFuzzer uses: *(i)* a fuzzer as a generator of candidate assertions derived from a grammar that is automatically obtained from the class definition; *(ii)* a dynamic invariant detector –Daikon– to filter out assertions invalidated by a test suite; and *(iii)* a mutation-based mechanism to cluster and rank assertions, so that similar constraints are grouped and then the stronger prioritized. Fuzzing, traditionally used to efficiently produce structured random data for testing, has two key advantages in this context: (1) it eliminates the need of developers to manually define candidate assertions and (2) it enables developers to straightforwardly adapt the language of assertions by manipulating the fuzzing grammar, e.g., to include additional operators.

We evaluated our technique on a benchmark of 43 Java methods employed in the evaluation of the state-of-the-art techniques GAssert and EvoSpex. In our evaluation, we used the same benchmarks from the evaluation of GAssert and EvoSpex, carefully studied the subjects, and manually produced corresponding "ground truth" assertions capturing the intended behavior of the subjects. We then used this ground truth to accurately assess precision and recall of SpecFuzzer, GAssert, and EvoSpex. Our results show that SpecFuzzer can easily support a more expressive assertion language, over which is more effective than GAssert and EvoSpex in inferring specifications, according to standard performance metrics. More precisely, SpecFuzzer was able to express ~45% more assertions in the ground truth than these tools. Also, SpecFuzzer was able to detect 75% of all assertions in the ground truth, showing a better overall performance compared to previous techniques. The results we obtained provide initial, yet strong evidence that SpecFuzzer is effective.

This work was published at the *44th International Conference on Software Engineering (ICSE 2022)* held in Pittsburgh, PA, USA on 22-27 May 2022.