

**Review of Steinhöfel D. and Zeller A.  
Language-Based Software Testing.  
*Commun. ACM* 67, 4 (2024). 80–84; 10.1145/3631520**

T.H. Tse,  
Department of Computer Science,  
The University of Hong Kong,  
Pokfulam, Hong Kong

Testing remains as the prime technique to check whether software satisfies the specified requirement. Owing to the high volume of potential inputs and expected outputs in very large systems, the process is laborious and error-prone if conducted manually. To meet the challenge, this article proposes the use of bots, or automatic robot processes, to test the systems continuously 24/7.

However, two hurdles prevail. The first is the huge number of potential inputs. Random testing is an obvious choice, but the diversity still needs human judgement. Second, we assume the existence of a test oracle that verifies the correctness of the outputs. Again, human intervention is often needed.

The present authors propose an excellent solution to the problem by (a) specifying the grammar and then (b) using their input specification language (ISLa) to obtain the properties of the components. They support their methodology via various techniques such as program synthesis, dynamic invariants, and explainable artificial intelligence (AI). They provide a comprehensive solution to the two common issues in software testing: the completeness of input coverage and the presence of undetected failures. Thus, the article is of great importance to novice testers for the fundamental concepts and to experienced testers for the detailed treatment.

I have a modest suggestion: The authors may consider the work of Tsong Yueh Chen, who has just been selected for the ACM SIGSOFT Outstanding Research Award 2024. His work on adaptive random testing (ART) proves that no testing method can reveal failures using less than half the test cases required by random testing, and ART is close to the optimum [1]. His work on metamorphic testing (MT) [2] shows that we may alleviate the oracle problem by comparing the inputs and outputs of *multiple* executions of the same software. Alastair F. Donaldson and team from Imperial College London have developed metamorphic fuzzers to uncover numerous bugs from compilers. In short, ART and MT together may ease the workload of the bots.

## References

- [1] Chen, T.Y. and Merkel, R. An upper bound on software testing effectiveness. *ACM Trans. Softw. Eng. Methodol.* 17, 3 (June 2008): 16:1–16:27; 10.1145/1363102.1363107.
- [2] Chen, T.Y. and Tse, T.H. New visions on metamorphic testing after a quarter of a century of inception. In *Ideas, Visions and Reflections Track, Proc. ACM Joint Eur. Softw. Eng. Conf. and Symp. Found. Softw. Eng.* (Athens, Greece, 2021), ACM, 1487–1490; 10.1145/3468264.3473136.