12 February 2025 · CC-BY 4.0

Qeios

Peer Review

Review of: "Finding Missed Code Size Optimizations in Compilers using LLMs"

Yu Wang[1]

[1]. Nanjing University, China

This paper combines large language models with a series of differential testing strategies, making differential testing suitable for identifying code size optimizations missed by compilers.
The method is simple, effective, clear, easy to understand, and easy to transplant to other languages.

But I have the following concerns:

1. What are the main challenges addressed by the differential testing strategy proposed in the paper? Compared with other differential testing work, the novelty might not be very significant.

2. At the end of Section 5, the paper contrasts with two recent works on identifying compiler-missed optimizations and concludes: “Compared to both these works, our approach is not language specific, requires no instrumentation of programs, and is the first work to use machine learning to generate code rather than handcrafted rules." However, these two works mainly contribute to the strategy of differential testing rather than code generation, so it seems inappropriate to compare with these two works in terms of code generation. In addition, the approach used in these two works (using "dead code markers") is also language-agnostic.

3. Missing some related work, such as [1].

[1] Ou, Xianfei, et al. "The Mutators Reloaded: Fuzzing Compilers with Large Language Model Generated Mutation Operators." ASPLOS, 2024.

Declarations

Potential competing interests: No potential competing interests to declare.

doi.org/10.32388/TPZ2HL