**Title:** Automatic Detection of Semantic Conflicts in Merge Commits via LLMs (e.g, GPT-4)

**Author:** Duarte Guedes Sardão

**Supervision:** José Campos

**Date:** 23 November 2023

## Abstract

In collaborative software development, parallel work done by several different branches often has to be merged. Due to the differences of work done in the difference branches, often conflicts arise. Some conflicts are easy to detect and rectify, such as textual merge conflictts, where different developers have altered the same line in different branches: most version control systems can detected that the same line has been changed and urge the merger to decided on a solution, for example, by keeping one change and discarding the other.

Among merge conflicts, semantic merges arise as particularly difficult to resolve, as they avoid detection by version control systems and lacking syntactic errors, compiled successfully. An example of a semantic conflict can be seen in a class such as Point, containing a method to calculate the distance to another Point: in a scenario where one developer in branch A changes the distance calculation (from euclidean to manhattan, for example), while another calls the distance method for a movement function in branch B, we find that upon a merge while no errors are raised, the code exhibits altered behaviour from the original branches: specifically, the movement will be different from what was developed in branch B.

We seek to explore how the emerging field of Large Language Models can provide a breaktrough in our ability to test for the altered, introduced and lost behaviours arising from semantic conflicts. Specifically, building upon previous work that identifies probable conflicts and generates outputs, on a domain-specific language, we analyse the ability of ChatGPT to generate appropriate unit tests that highlight the presence of semantic conflicts, given appropriate prompting.

**Keywords:** Merge Conflicts, Semantic Conflicts, Large Language Models, Software Testing, ChatGPT

**ACM Classification:** Software and its engineering -> Software creation and management -> Software verification and validation

Software and its engineering -> Software notations and tools -> Context Specific Languages -> Domain Specific Languages

Computing Methodologies -> Machine Learning

## References

[1] T. Mens. A state-of-the-art survey on software merging, 2002. Available at `https://ieeexplore.ieee.org/document/1000449`.

[2] Nuno Castanho. Semantic conflicts in version control systems, 2021. Available at `https://repositorio.ul.pt/handle/10451/50658`.

[3] Junjie Wang, Chunyang Chen Yuchao Huang, Zhe Liu, Song Wang, and Qing Wang. Software testing with large language model: Survey, landscape, and vision, 2023. Available at https://arxiv.org/abs/2307.07221.

[4] Vitor Guilherme and Auri Vincenzi. An initial investigation of chatgpt unit test generation capability, 2023. Available at https://dl.acm.org/doi/10.1145/3624032.3624035.