

**Name:** Ying Zhang

**Date:** February 19, 2019.

**Paper:** Huang, Kaifeng, Bihuan Chen, Xin Peng, Daihong Zhou, Ying Wang, Yang Liu, and Wenyun Zhao, “CIDiff: generating concise linked code differences”.

**Summary:** (Problem) The exists code the code change summarization methods are too coarse-grained; the code differencing methods are too fine-grained and the relationship among code changes are missing, which makes hard for code change analyzing and understanding. (Solution) Huang et. al [1] proposed and implemented a novel code differencing method named CIDiff; this approach is able to generate the concise code differencing, build code change link between differencing based on pre-defined links. (Meaning) This approach may help developers quickly understand the code differencing to resolve the merge conflict; provide a useful tool for future research such as code differencing analysis.

**Strengths:** Generating the concise code differencing (human understandable) is more useful for developers. The idea for grouping the in or higher statement-level code differencing then provide the code differencing is novel.

**Weaknesses:** For the evaluation part, 1) this paper didn't offer recall of the tool but just precision. 2) decide the tool is useful based on 10 participants who are graduate students from their school. (personally, I think the result in this part is not objective.)

**Analysis I:** In the accuracy evaluation part, this paper made conclusion purely based on precision, which is not convinced. For example, if the tool could only distinguish 10% of the code differencing, even 99% of the concise code differencing is correct, we can not conclude it perform well. Therefore, I think they should provide the recall in table 2.

**Analysis II:** According to the snapshot of their visualization tool, the code differencing information provided to the developer is still hard for the human to read. The information primarily provides the change of the code, but lack of the motivation for code changing. For example, in Figure 5, the message "addMethodDeclaration" didn't provide any detail information about why make this change, it is still hard for developers to make code merging based on this information. In my opinion, it should allow programmers to add comments to the message or extract the related comment when showing the message. That will help the programmer to understand the changes better.

## References

- [1] J.-R. Falleri, F. Morandat, X. Blanc, M. Martinez, and M. Monperrus. Fine-grained and accurate source code differencing. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*, pages 313–324. ACM, 2014.