Repairing functional dependencies with one-path Chasing[1]

Toon Boeckling [†] and Antoon Bronselaer [†]

[†] DDCM lab, Department of Telecommunications and Information Processing, Ghent University, Ghent, Belgium
{toon.boeckling,antoon.bronselaer}@ugent.be

The repair problem for functional dependencies (FDs) is the problem where an input database needs to be modified such that all FDs are satisfied and the difference with the original database is minimal. The output database is then called an optimal repair. If the output database is constructed by updating individual values, finding such an optimal repair is NP-hard [1]. A well-known approach to find approximations of optimal repairs is the Llunatic Chase [2]. This algorithm constructs a Chase tree in which each internal node resolves violations of one FD and leaf nodes represent repairs. By controlling the branching factor of the Chase tree, one controls the trade-off between repair quality and computational efficiency. We explore an extreme variant of this idea in which the Chase tree has only one path. To construct this path, we first create an ordered partition of attributes such that classes can be repaired sequentially. This latter means that once classes are repaired, attributes from that class do not change anymore. We then repair each class one by one and do so by fixing the order in which dependencies are repaired. This principle is called priority repairing and we provide a simple heuristic to determine priority. The techniques for attribute partitioning and priority repair are combined in the Swipe algorithm. An empirical study on four real-life data sets shows that Swipe is one to three orders of magnitude faster than multi-sequence Chase-based approaches (Llunatic) and learning-based approaches (HoloClean). At the same time, the quality of repairs is comparable or better. A scalability analysis of the algorithm shows that Swipe scales linearly in terms of an increasing number of tuples. In general, it scales quadratic in terms of an increasing number of FDs. However, for unary FDs, it scales linearly when violations of FDs are resolved by choice.

References

[1] Kolahi, S., and Lakshmanan, L. V. S. (2009). On approximating optimum repairs for functional dependency violations. *In Proceedings of the 12th International Conference on Database Theory*, 53-–62.

[2] Geerts, F., Mecca, G., Papotti, P., and Santoro, D. (2019). Cleaning data with Llunatic. *The VLDB Journal*, 29, 867-–892.

[1] Full version available at https://arxiv.org/abs/2403.19378