# **Documentation FunctionalDerpendencies Algorithm**

The FunctionalDerpendencies-Algorithm follows the FastFD approach to look for functional dependencies, creating stripped partitions, agree sets and the cover tree in the process.

In order to limit the columns and data in question the algorithm handles partitions occurring in the columns as bitsets and ignores all single occurrence partitions when creating the stripped partitions.

Although the FastFd algorithm relies on agree sets the simplified way to compute agree sets as it was presented in the paper did not provide convincing results beyond small integer datasets. Thus the FunctionalDerpendencies algorithm creates agree sets as proposed in the Dep-Miner algorithm. The set generation requires the retrieval of maximum sets before – this is done using the stripped partitions and used in the creation of the difference sets. Agree sets themselves are created by examining the stripped partitions again and creating equivalence classes out of them, then these equivalence classes are intersected to find the final agree set items. It is apparent that the creation of equivalence classes should be optimized in future revisions of the FunctionalDerpendencies algorithm – especially since the maximum set search uses a similar approach analyzing the stripped partitions.

The performance is in line with the expectations, with the typical limitations of set based algorithms for functional dependency discovery, as the FunctionalDerpendency algorithm runs out of memory for relations with many columns and is slower than e.g. fdep on the ncvoter dataset.

All tests were obtained using an Intel i5/4300U running at 1.9GHz with 8 GB of main memory and a Hynix solid state drive.

NULL values are treated as different values, since NULL may not be a value to rely or depend on.