Exercise 1
Unique Column Combinations

- Deadline: **Monday, 10.11.14**
- The admission to the exam requires *all* exercises to be solved.
- The exercises should be solved in teams of two students.
- The Metanome project is available at GitHub:
  `https://github.com/HPI-Information-Systems/Metanome`
- The datasets and supplemental material can be found at network drive S:
  `\\fs3\bbs\DPDC`
- The submission system can be found at:
  `https://www.dcl.hpi.uni-potsdam.de/submit/`
- To solve an exercise, please submit a zip file containing the following items:
  - **<algorithm_name>.jar**: An executable Metanome algorithm.
  - **<algorithm_name>.zip**: The algorithm's source code (maven project).
  - **<algorithm_name>_docu.pdf**: Short documentation of the algorithm.
  - **<algorithm_name>_pres.pptx/ppt/pdf**: Two slides presentation of the algorithm.

Task 1: Unique Column Combinations - A discovery algorithm

Write an algorithm that discovers *all minimal* unique column combinations on the given datasets. The rules for your implementation are as follows:

a) The algorithm discovers *exact* results, so no approximate or fuzzy results are allowed.

b) The algorithm is not allowed to use parallelization.

c) The algorithm implements the Metanome interface and is compatible with Metanome.

d) For the NULL semantic, assume NULL $\neq$ NULL.

You can re-implement an existing algorithm from literature or find your own algorithm. To test and evaluate the algorithm, use the datasets provided on the network share. Your algorithm should at least be able to process the WDC_planets dataset! Before submitting your algorithm, check that it correctly executes within Metanome!

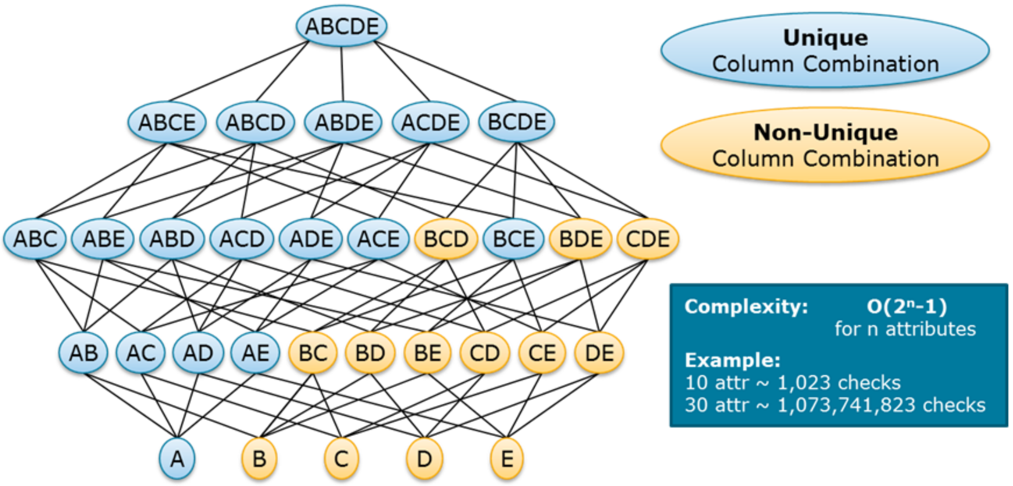

Abbildung 1: Unique Column Combinations.

BONUS TASK: If you like to dive deeper, you can try to parallelize your discovery algorithm. What effect does parallelization have on your pruning strategies? You can also try to make the search approximate or conditional. Which changes make your algorithm faster and which make it slower? If you made changes to your main algorithm, please submit them as a separate algorithm, e.g. <algorithm_name>_conditional.jar

Task 2: Documentation

Write a short (max one A4 page) documentation for your algorithm describing the algorithm that you implemented:

a) Describe the algorithm's basic idea. How does the algorithm cope with the complexity of the given task?

b) If you used an algorithm from literature, provide a reference to the according publication.

c) If you came up with an own approach, provide one or two arguments why it is or could be better than related algorithms.

d) If you algorithm implements an adaption or optimization of existing approaches, describe these briefly.

e) State if your algorithm (jar) should be published as Metanome algorithm providing you as the authors.

f) *If you solved a bonus task, please discuss your findings here as well.*

In the same document, answer the following questions:

a) How many unique column combinations did your algorithm find on the provided datasets?

b) How long did the discovery take on each dataset and what machine did you use?

c) Did you discover any limitations of your approach (e.g. runtime or memory consumption) that made computing a certain dataset impossible?

d) What is the conceptual differences between NULL $\neq$ NULL and NULL $=$ NULL and how does the choice of the NULL semantic influence the performance of your algorithm?

Task 3: Presentation

Prepare two slides for a short, 5 min presentation of your algorithm in the lecture. One slide about the algorithm and one slide about its performance. Here are some ideas for these slides:

a) Explain the idea of your approach and why you think it works well in comparison to others.

b) How did your algorithm perform on the given datasets?

c) Did you make any interesting observations?

d) What lets your algorithm crash?

e) How hard was it to implement your algorithm?

f) *This is also a good time to introduce an optimization or adaption of your algorithm.*

Note that each team will present its work once!