HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Data Profiling and Data Cleansing - Assignment 1

Unique Column Combinations

Group:
Christoph Oehlke christoph.oehlke@student.hpi.uni-potsdam.de
Markus Hinsche markus.hinsche@student.hpi.uni-potsdam.de

Bottom-up Checking Using PLIs

- Bottom-up checking
- Using PLIs for every column (ignore actual values)
- Building a column combination means intersecting all PLIs of column A with all PLIs of column B
- Building n-dimensional combinations:
  - intersect (n-1)-dimensional PLIs with 1-dimensional PLIs
  - Saves memory, as we can delete all PLIs from 2 to (n-2)

Example: **AB** -> ABC, ABD, ABE | **AC** -> ACD, ACE | **AD** -> ADE
BC -> BCD, BCE | **BD** -> BDE | **CD** -> CDE

- **Problem:** Search space grows exponentially…

Optimization: Max-unique-pruning

- Let **X**, **Y** be sets of columns.
- **uniques(X)** := number of uniques in X
- **uniques(Y)** := number of uniques in Y
- **{X,Y}** := combination of column sets X and Y

- When having built {X,Y} out of X and Y:
  - Check if **uniques({X,Y}) > max(uniques(X), uniques(Y))**
  - If **false**: Drop {X,Y} from memory

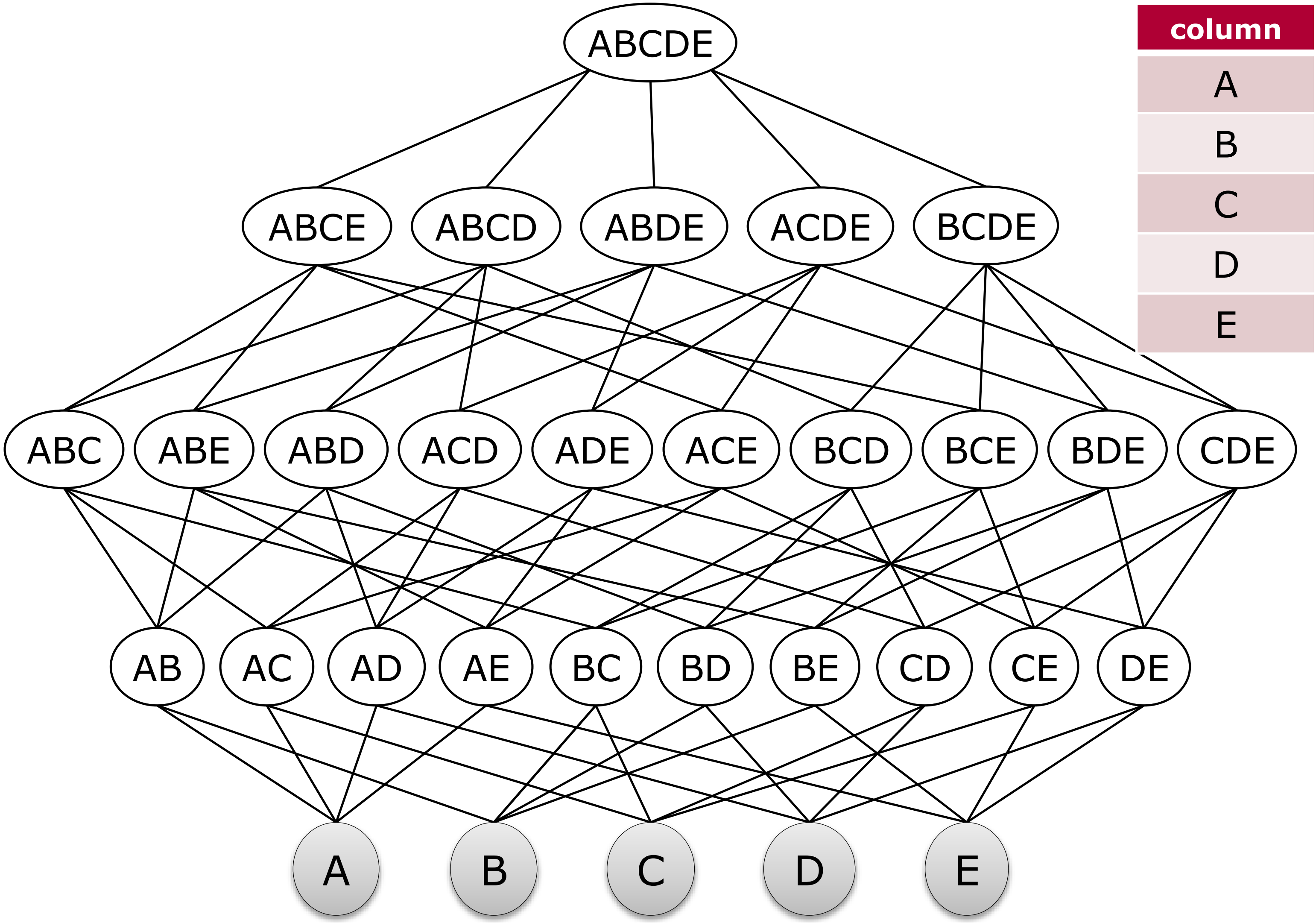
- Is an „aggressive" pruning technique:
  - Massively reduces numbers of combinations to check
  - But leads to loss of some unique combinations

Initial Column Pruning

- Initial pruning of ‚bad‘ columns, based on number of uniques
  - For example, ignore all columns having <= 20% uniques
  - Operating on ~25-60 columns instead of 223 (threshold 1-10%)

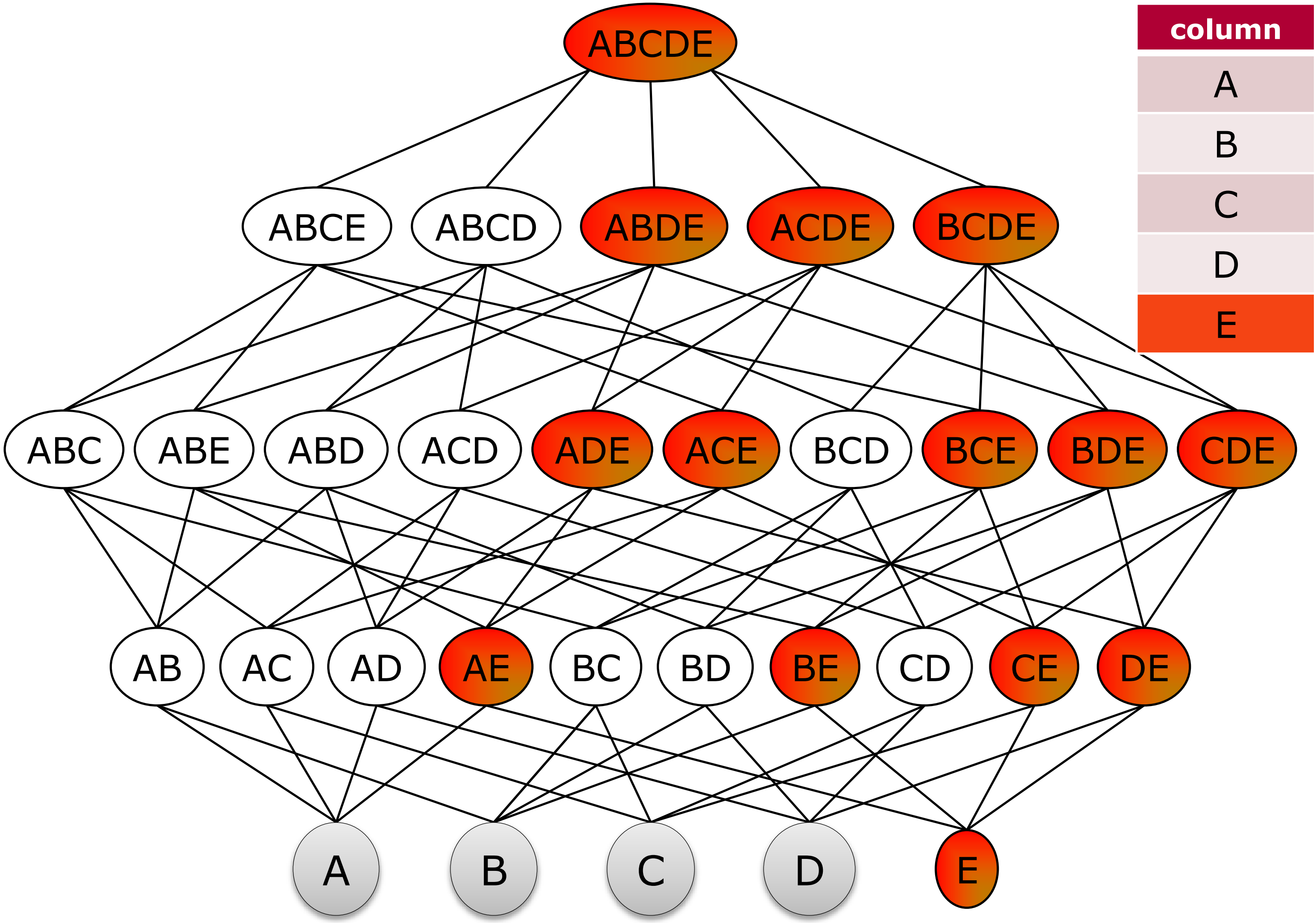
A	B	C	D	E	F
U	1	x	1	⊥	a
V	2	y	2	⊥	b
V	3	z	5	⊥	c
W	3	⊥	5	a	d
X	⊥	⊥	5	⊥	e
⬇	⬇	⬇	⬇	⬇	⬇
4/5 unique	3/5 unique	3/5 unique	2/5 unique	1/5 unique	5/5 unique
				⬇	⬇
				DROP	DROP

Initial Pruning



column	uniques
A	80%
B	60%
C	60%
D	40%
E	20%

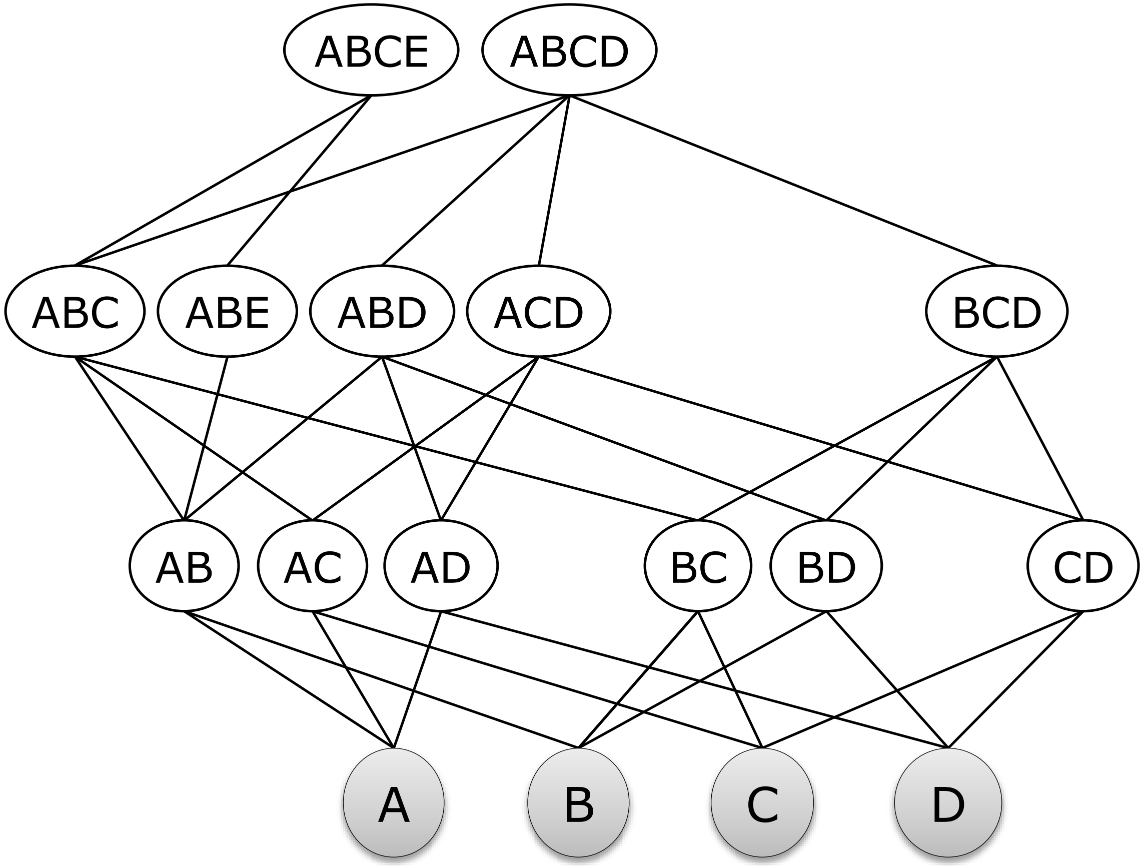
Initial Pruning



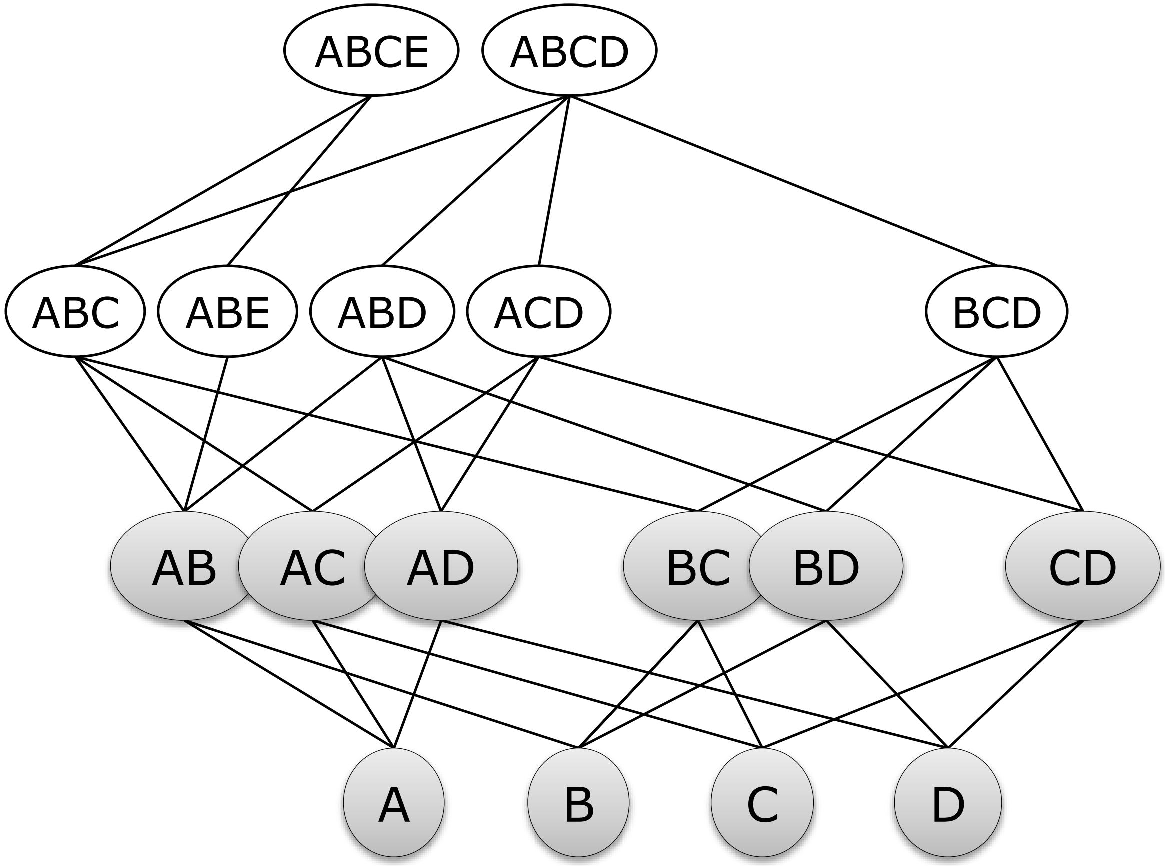
column	uniques
A	80%
B	60%
C	60%
D	40%
E	20%

Initial Pruning

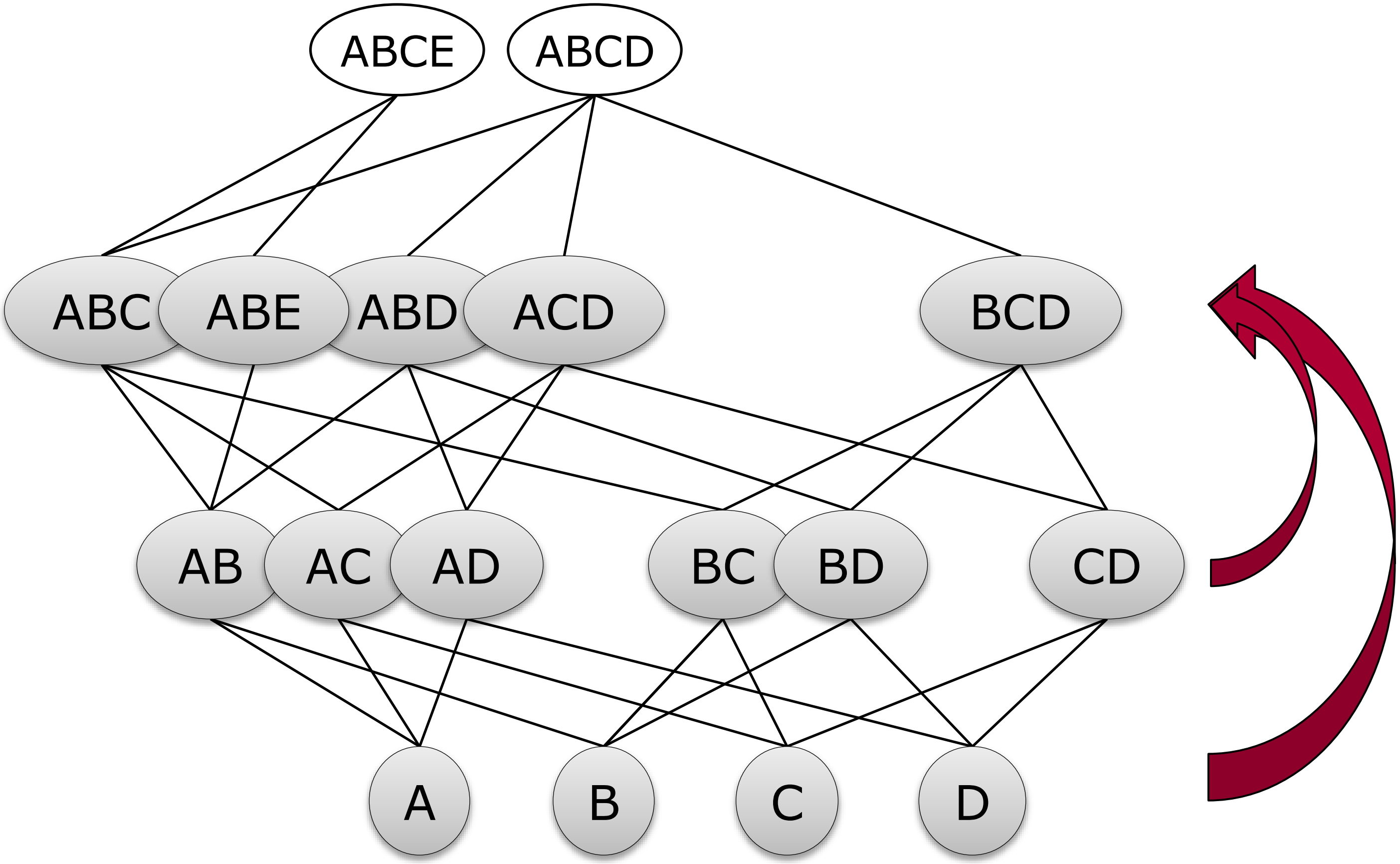
column	uniques
A	80%
B	60%
C	60%
D	40%
E	20%



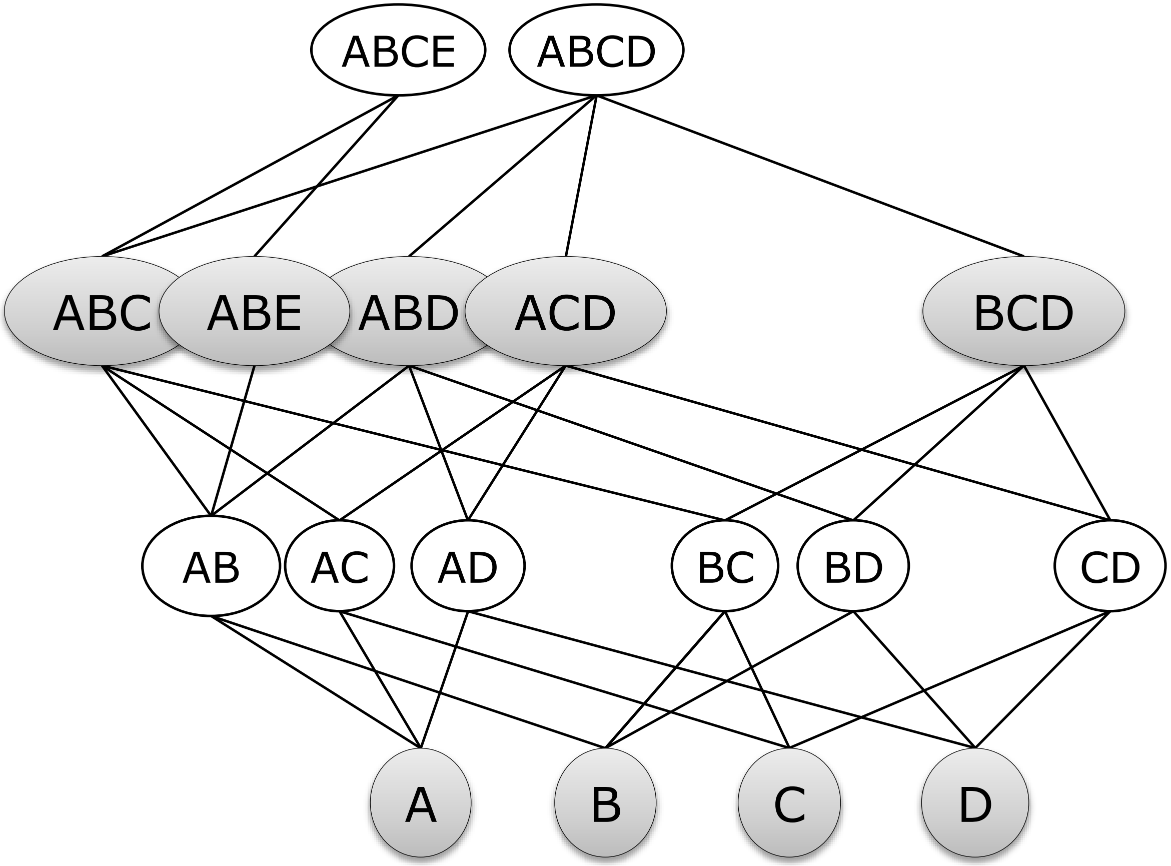
Building Column Combinations



Building Column Combinations

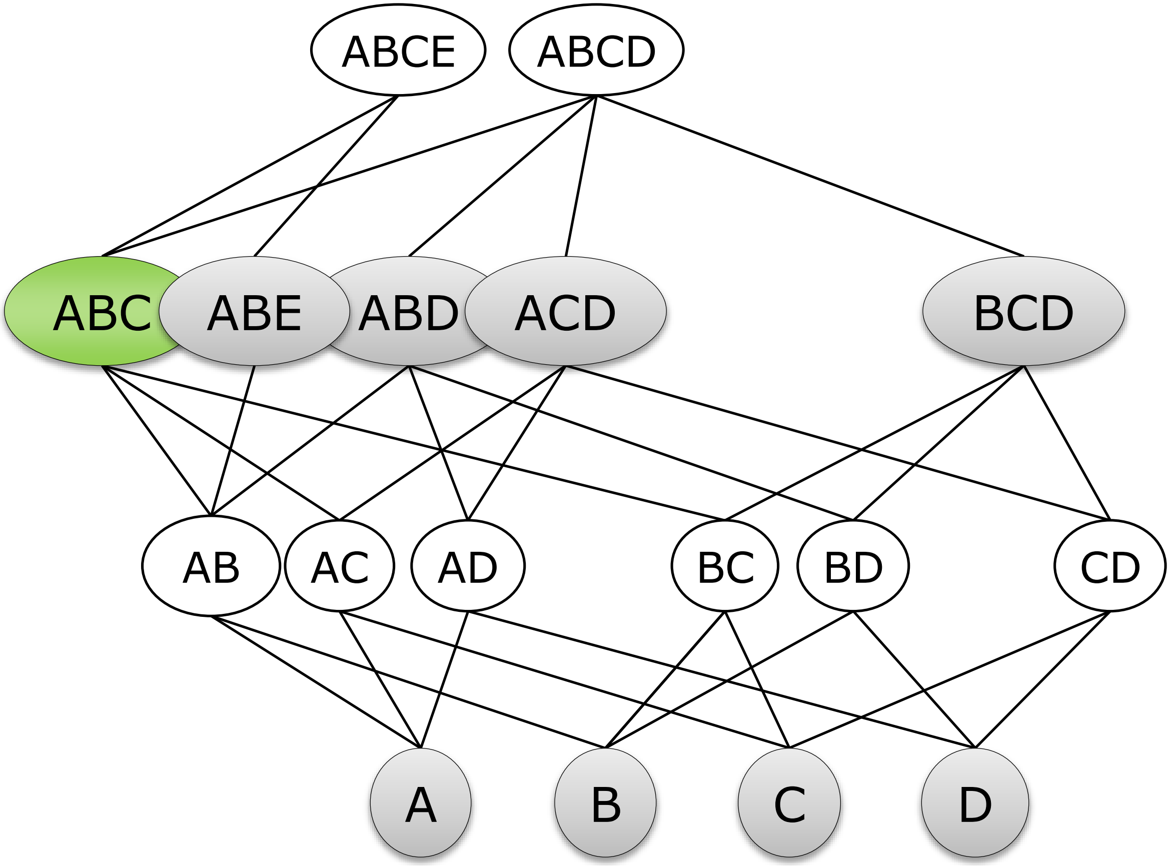


Building Column Combinations



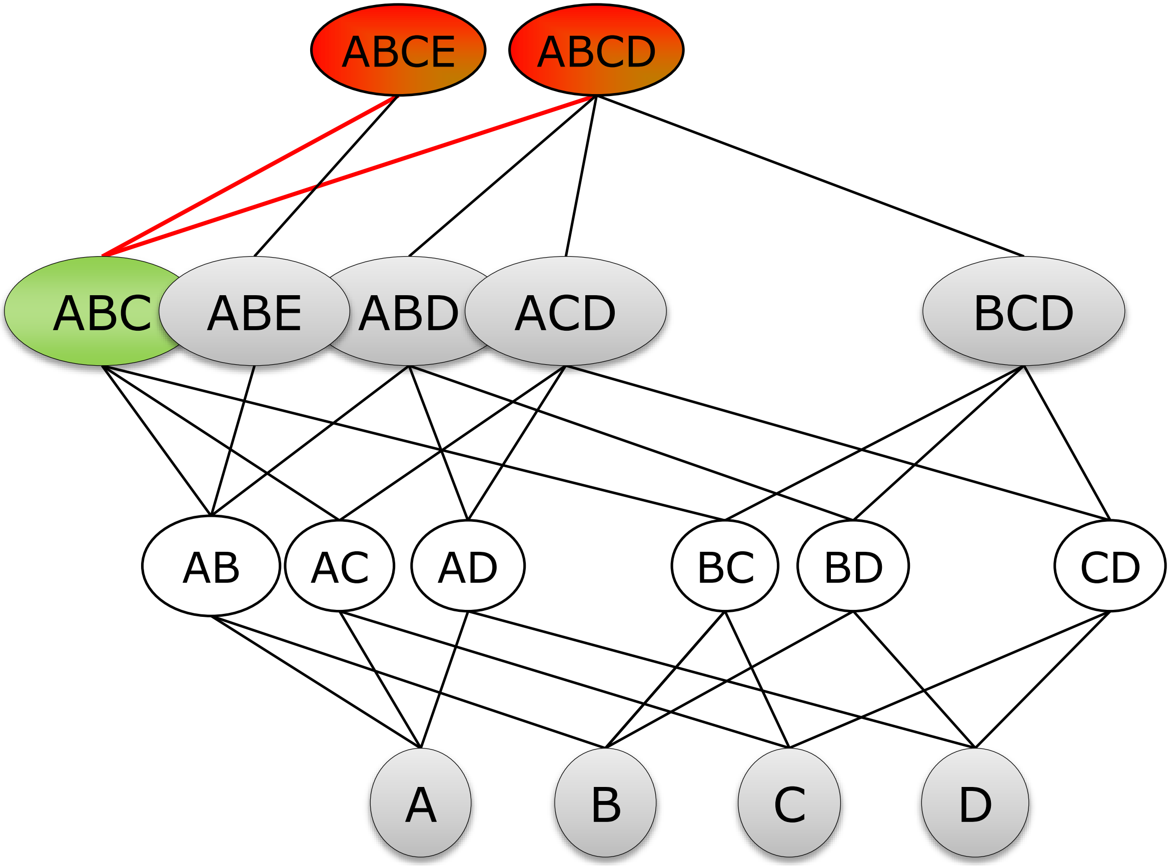
=> Remove from memory

Building Column Combinations



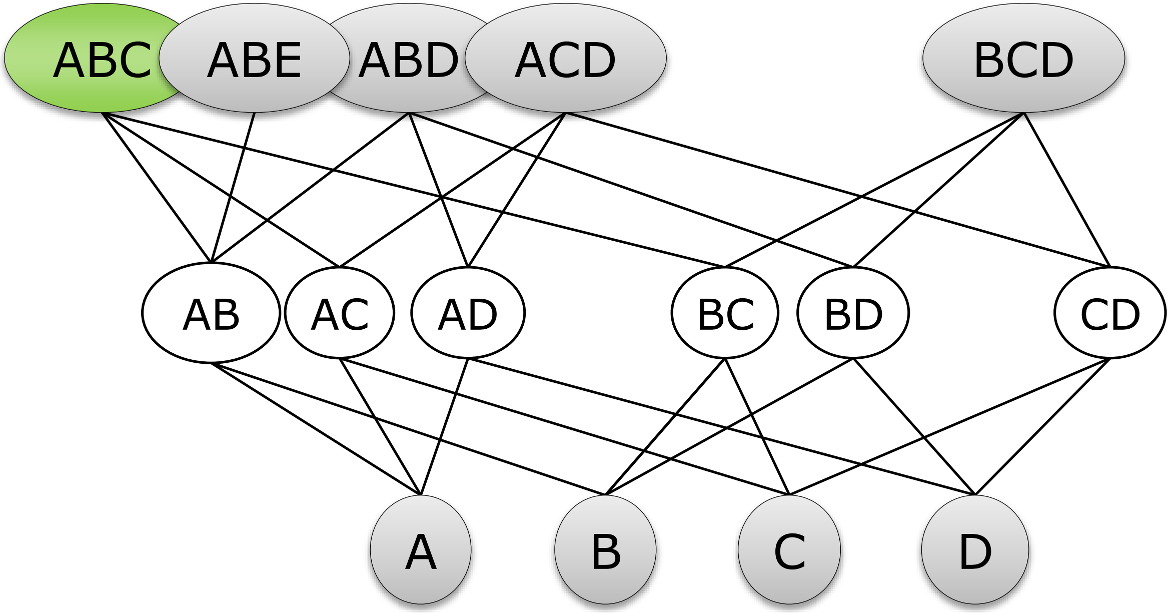
Found unique!

Building Column Combinations



Found unique!

Building Column Combinations



Found unique!

Results

DPDCAssignment1ChristophOehlkeMarkusHinsche.tsv

0
1

Outlook

- Key problem: finding a good value for the threshold
- Trade-off:
  - Low threshold -> less initial pruning -> high complexity
  - High threshold -> aggressive pruning -> uniques missing

- Possible improvement:
  - Split large dataset into n smaller datasets
  - Find unique combinations on each of the n subsets
  - Final step: check which of these unique combinations are also valid for the large dataset