A method to find functional dependencies through refutations and duality of hypergraphs

J. Fuentes, P. Sáez, G. Gutiérrez and ID Scherson

Universidad del Bío-Bío, Chile
University of California, Irvine, USA

November 17, 2015

Outline

Some definitions

Functional Dependency:

Let R be a relation with a set A of attributes and let r be an instance of R. A *functional dependency* (FD) is an expression of the form $X \rightarrow Y$, where $X \subseteq A$ and $Y \in A$. X is called the *determinant set* and Y is called the *dependent attribute*.

The dependency is *valid* in the instance r if and only if for every pair of rows (tuples) $t, u \in r$, whenever $t[B] = u[B]$ for all $B \in X$, it is also the case that $t[Y] = u[Y]$.

Some definitions

Functional Dependency:

Example

Tuple ID	A	B	E	C	D
1	1	a	2	$	Flower
2	1	A	2		Tulip
3	2	A	0	$	Daffodil
4	2	A	0	$	Flower
5	2	B	0		Lily
6	3	B	1	$	Orchid
7	3	C	1		Flower
8	3	C	1	#	Rose

Some definitions

Functional Dependency:

Example

Tuple ID	A	B	E	C	D
1	1	a	2	$	Flower
2	1	A	2		Tulip
3	2	A	0	$	Daffodil
4	2	A	0	$	Flower
5	2	B	0		Lily
6	3	B	1	$	Orchid
7	3	C	1		Flower
8	3	C	1	#	Rose

Some definitions

Functional Dependency:

Example

Tuple ID	A	B	E	C	D
1	1	a	2	$	Flower
2	1	A	2		Tulip
3	2	A	0	$	Daffodil
4	2	A	0	$	Flower
5	2	B	0		Lily
6	3	B	1	$	Orchid
7	3	C	1		Flower
8	3	C	1	#	Rose

FD: $A \rightarrow E$

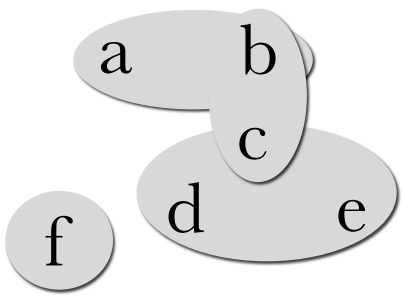
Some definitions

Hypergraph:
A hypergraph is defined as a generalized graph $H = (A, E)$, where A is a finite set of vertexes and $E \subseteq \mathcal{P}(A)$ is a set of *hyperedges* ($\mathcal{P}(C)$ is the set of subsets of C).

Some definitions

Hypergraph:
A hypergraph is defined as a generalized graph $H = (A, E)$, where A is a finite set of vertexes and $E \subseteq \mathcal{P}(A)$ is a set of *hyperedges* ($\mathcal{P}(C)$ is the set of subsets of C).

Example: Let $A = \{a, b, c, d, e, f\}$, and let H be the set of hyperedges $\{\{a, b\}, \{b, c\}, \{c, d, e\}, \{f\}\}$.



$$H = \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Some definitions

Transversal of a hypergraph: is a hyperedge that intersects all hyperedges in H .

A transversal X is called minimal if no proper subset of X is a transversal.

The operator λ generates precisely the set of minimal transversals of an hypergraph.

Some definitions

Transversal of a hypergraph: is a hyperedge that intersects all hyperedges in H .

A transversal X is called minimal if no proper subset of X is a transversal.

The operator λ generates precisely the set of minimal transversals of an hypergraph. Obs: This operator is a NP problem.

Example

$$H = \begin{array}{c} \begin{matrix} a & b & c \end{matrix} \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array} \qquad \lambda(H) = \begin{array}{c} \begin{matrix} a & b & c \end{matrix} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array}$$

Problem Statement

Given a relation R and an instance r of R, find all non-trivial FDs that are valid in r

Problem Statement

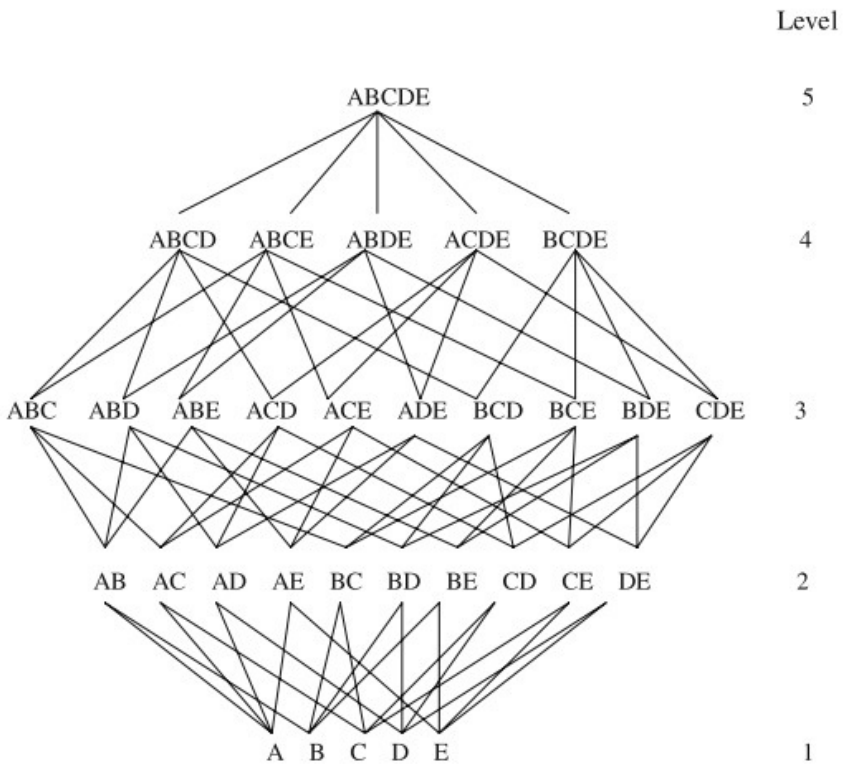
Given a relation R and an instance r of R, find all non-trivial FDs that are valid in r

Several algorithms have been proposed to solve this problem, but their performance decreases when the number of attributes in a relation is big (for instance, more than 25 or 30 attributes).
The reason is that the size of the boolean lattice grows exponentially with the number of attributes.

Problem Statement

For example, a relation with the set of attributes $\{A, B, C, D, E, F\}$ and the we are interested in finding the FDs $X \rightarrow F$.

The following combinations of attributes for X must be checked in r.



Method to find Functional Dependencies

Our method is based on finding first a set of refutations in r, avoiding the boolean lattice.

A **Refutation** is represented as $X \nrightarrow Y$ where $X \subseteq A$ and $Y \in A$ holds if and only if there exist two tuples t and u in r such that $(\forall B \in X)\ t[B] = u[B] \wedge t[Y] \neq u[Y]$.

A refutation X is maximal if no proper superset of X is a refutation. So a set of maximal refutations is where there is no refutation that is a subset of another refutation.

Method to find Functional Dependencies

The method has 4 main steps:

1. Obtain the set of maximal refutations from r.
   If we have the refutations $A, B, C \nrightarrow D$ and $A, B \nrightarrow$, we only consider $A, B, C \nrightarrow D$.
2. Generate hypergraph H as a boolean matrix with the maximal refutations that were obtained in the previous step.
3. Compute the complement of each maximal refutation in H, obtaining $H' = \{A \setminus X_1, A \setminus X_2, \ldots, A \setminus X_k\}$.
4. Compute minimal transversals to matrix H'. Here we obtain the desired set of minimal FDs.

Example

Given the set of attributes $A = \{1, 2, 3, 4, 5, 6\}$ and the instance r shown below, we are going to produce the set of FDs that have 6 as its dependent attribute, $X \rightarrow 6$.

Tuple ID	**1**	**2**	**3**	**4**	**5**	**6**
1	a_1	b_3	c_2	d_1	e_4	f_1
2	a_1	b_3	c_3	d_3	e_1	f_2
3	a_2	b_3	c_5	d_1	e_5	f_4
4	a_3	b_3	c_2	d_3	e_3	f_1
5	a_4	b_2	c_2	d_8	e_2	f_1
6	a_5	b_4	c_4	d_1	e_3	f_1
7	a_1	b_1	c_3	d_7	e_6	f_2

Example

1. Obtain the set of maximal refutations $X \nrightarrow 6$ from r.

Tuple ID	**1**	**2**	**3**	**4**	**5**	**6**
1	a_1	b_3	c_2	d_1	e_4	f_1
2	a_1	b_3	c_3	d_3	e_1	f_2
3	a_2	b_3	c_5	d_1	e_5	f_4
4	a_3	b_3	c_2	d_3	e_3	f_1
5	a_4	b_2	c_2	d_8	e_2	f_1
6	a_5	b_4	c_4	d_1	e_3	f_1
7	a_1	b_1	c_3	d_7	e_6	f_2

Example

1. Obtain the set of maximal refutations $X \nrightarrow 6$ from r. Look for pairs with different values on attribute 6.

Tuple ID	**1**	**2**	**3**	**4**	**5**	**6**
1	a_1	b_3	c_2	d_1	e_4	f_1
2	a_1	b_3	c_3	d_3	e_1	f_2
3	a_2	b_3	c_5	d_1	e_5	f_4
4	a_3	b_3	c_2	d_3	e_3	f_1
5	a_4	b_2	c_2	d_8	e_2	f_1
6	a_5	b_4	c_4	d_1	e_3	f_1
7	a_1	b_1	c_3	d_7	e_6	f_2

Example

1. Obtain the set of maximal refutations $X \nrightarrow 6$ from r. Look for similar values on the rest of attributes.

Tuple ID	1	2	3	4	5	6
1	a_1	b_3	c_2	d_1	e_4	f_1
2	a_1	b_3	c_3	d_3	e_1	f_2
3	a_2	b_3	c_5	d_1	e_5	f_4
4	a_3	b_3	c_2	d_3	e_3	f_1
5	a_4	b_2	c_2	d_8	e_2	f_1
6	a_5	b_4	c_4	d_1	e_3	f_1
7	a_1	b_1	c_3	d_7	e_6	f_2

Refutations: $1 \nrightarrow 6$

Example

1. Obtain the set of maximal refutations $X \nrightarrow 6$ from r. Look for similar values on the rest of attributes.

Tuple ID	1	2	3	4	5	6
1	a_1	b_3	c_2	d_1	e_4	f_1
2	a_1	b_3	c_3	d_3	e_1	f_2
3	a_2	b_3	c_5	d_1	e_5	f_4
4	a_3	b_3	c_2	d_3	e_3	f_1
5	a_4	b_2	c_2	d_8	e_2	f_1
6	a_5	b_4	c_4	d_1	e_3	f_1
7	a_1	b_1	c_3	d_7	e_6	f_2

Refutations: $1 \nrightarrow 6$, $2 \nrightarrow 6$, $\{1, 2\} \nrightarrow 6$

Example

1. Obtain the set of maximal refutations $X \nrightarrow 6$ from r. Keep the maximal refutations.

Tuple ID	1	2	3	4	5	6
1	a_1	b_3	c_2	d_1	e_4	f_1
2	a_1	b_3	c_3	d_3	e_1	f_2
3	a_2	b_3	c_5	d_1	e_5	f_4
4	a_3	b_3	c_2	d_3	e_3	f_1
5	a_4	b_2	c_2	d_8	e_2	f_1
6	a_5	b_4	c_4	d_1	e_3	f_1
7	a_1	b_1	c_3	d_7	e_6	f_2

Refutations: ~~$1 \nrightarrow 6$~~, ~~$2 \nrightarrow 6$~~, $\{1, 2\} \nrightarrow 6$

Example

1. Obtain the set of maximal refutations from r.

Tuple ID	**1**	**2**	**3**	**4**	**5**	**6**
1	a_1	b_3	c_2	d_1	e_4	f_1
2	a_1	b_3	c_3	d_3	e_1	f_2
3	a_2	b_3	c_5	d_1	e_5	f_4
4	a_3	b_3	c_2	d_3	e_3	f_1
5	a_4	b_2	c_2	d_8	e_2	f_1
6	a_5	b_4	c_4	d_1	e_3	f_1
7	a_1	b_1	c_3	d_7	e_6	f_2

Refutations: $\{1, 2\} \nrightarrow 6$

Example

1. Obtain the set of maximal refutations from r.

Tuple ID	1	2	3	4	5	6
1	a_1	b_3	c_2	d_1	e_4	f_1
2	a_1	b_3	c_3	d_3	e_1	f_2
3	a_2	b_3	c_5	d_1	e_5	f_4
4	a_3	b_3	c_2	d_3	e_3	f_1
5	a_4	b_2	c_2	d_8	e_2	f_1
6	a_5	b_4	c_4	d_1	e_3	f_1
7	a_1	b_1	c_3	d_7	e_6	f_2

Refutations: $\{1,2\} \nrightarrow 6$
$\{2,4\} \nrightarrow 6$

Example

2. Generate hypergraph H as a boolean matrix with the maximal refutations that were obtained in the previous step.

Refutations: $\{1,2\} \nrightarrow 6$
$\{2,4\} \nrightarrow 6$

$$H = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Example

2. Generate hypergraph H as a boolean matrix with the maximal refutations that were obtained in the previous step.

Refutations: $\{1,2\} \nrightarrow 6$
$\{2,4\} \nrightarrow 6$

$$H = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Property of H:

- There is no hyperedge in H that is subset/superset of another hyperedge.

Example

3. Compute the complement of each maximal refutation in H, obtaining $H' = \{A \setminus X_1, A \setminus X_2, \ldots, A \setminus X_k\}$.

$$H = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array} \qquad H' = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

Example

4. Compute minimal transversals to matrix H'. Here we obtain the desired set of minimal FDs.

$$H' = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array} \qquad \lambda(H') = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Example

4. Compute minimal transversals to matrix H'. Here we obtain the desired set of minimal FDs.

$$H' = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array} \qquad \lambda(H') = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

minimal FDs: $\{\ \{1,4\} \to 6, \quad \{3\} \to 6, \quad \{5\} \to 6\ \}$

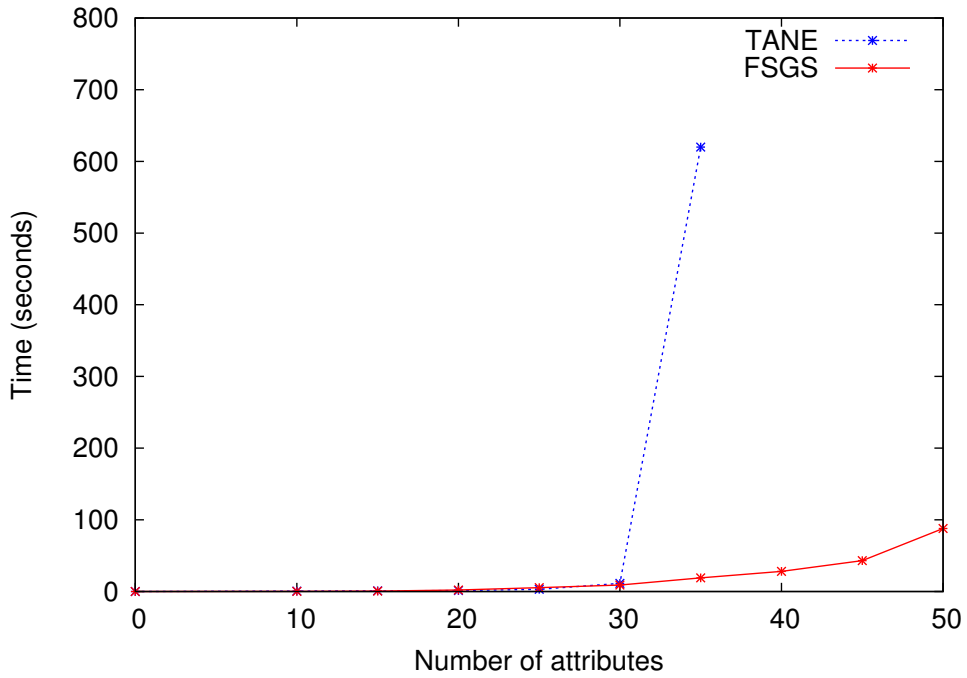
Example

Final result

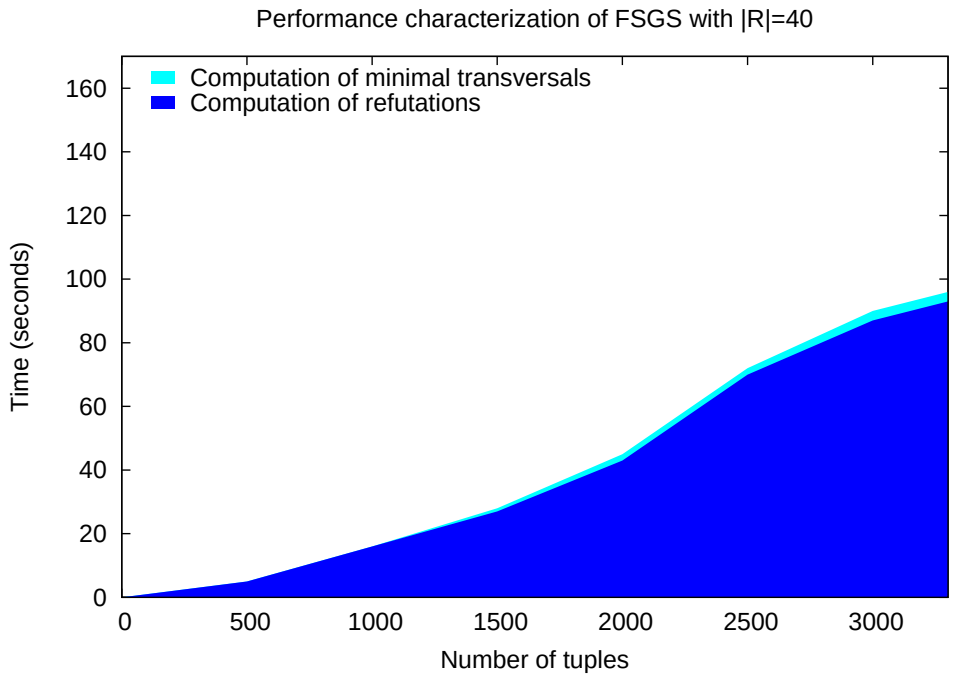
minimal FDs: $\{\ \{1,4\} \to 6,\quad \{3\} \to 6,\quad \{5\} \to 6\ \}$

Tuple ID	**1**	**2**	**3**	**4**	**5**	**6**
1	a_1	b_3	c_2	d_1	e_4	f_1
2	a_1	b_3	c_3	d_3	e_1	f_2
3	a_2	b_3	c_5	d_1	e_5	f_4
4	a_3	b_3	c_2	d_3	e_3	f_1
5	a_4	b_2	c_2	d_8	e_2	f_1
6	a_5	b_4	c_4	d_1	e_3	f_1
7	a_1	b_1	c_3	d_7	e_6	f_2

Experimental Results



Experimental Results



Worst case

The proposed method uses an auxiliary data structure to keep the maximal refutations (matrix H).
Let k be the number of attributes and p the number of attributes participating in a refutation (set as 1's in the matrix H).
The worst case occurs when:

- Each refutation is maximal in matrix H
- The number of ones p in each refutation is $k/2$
  E.g. for $k = 4$

$$H = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

Worst case

The worst case occurs when:

- Each refutation is maximal in matrix H
- The number of 1's p in each refutation is $k/2$

The maximum number of maximal refutation given by:

$$\frac{k!}{p!(k-p)!}$$

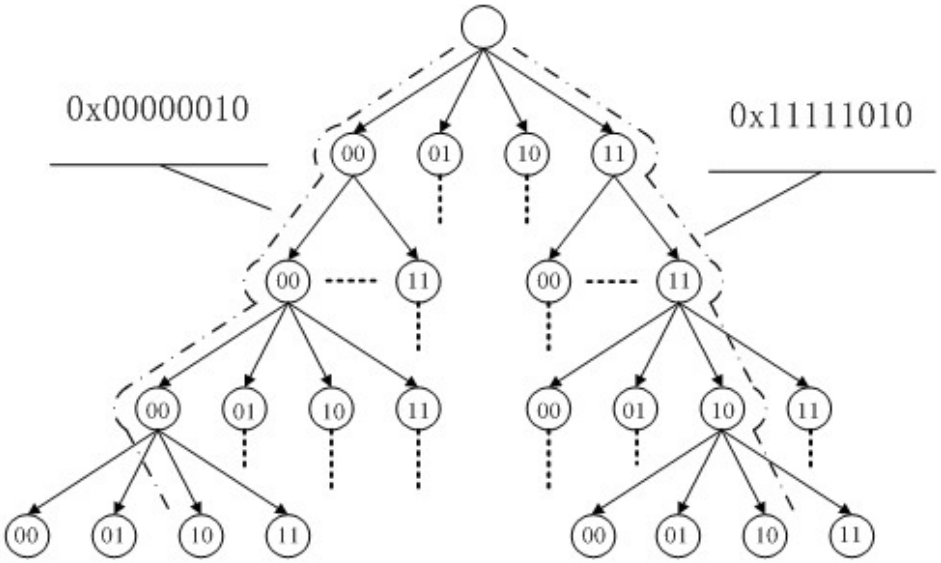
To get this worst case, the previous step of finding refutation through the relation, needs to have the following characteristics:

- Each attribute with a big cardinality to produce the maximal refutations
- The domain for each attribute must be large enough to achieve the previous condition

Improvements to deal with the worst case

Some improvements can be done in order to get a better performance.

- Compress the matrix H into a new data structure: radix tree
  - Each node works as a block with b bits
  - Refutations are represented by the path from the root to the leaves



Improvements to deal with the worst case

Some improvements can be done in order to get a better performance.

- Parallelize the task of finding refutation over the relation. Here the relation can be divided and each tuple is compared with all the tuples in each subset.

Conclusions

- The method finds the set of minimal FDs properly.
- The method has better performance than traditional algorithms.
- The step of finding refutations (whose running time is quadratic) takes longer than computation of minimal transversals (whose running time is subexponential).

References

Mannila, H. and Rih, K.-J. (1994)

Algorithms for inferring functional dependencies from relations

Data Knowl. Eng., 12, 8399.

Huhtala, Y., Karkkainen, J., Porkka, P. and Toivonen, H. (1999)

Tane: an efficient algorithm for discovering functional and approximate dependencies.

Comput. J., 42, 100

Berge, C. (1989)

Hypergraphs.

Elsevier Science Publishers B.V., Amsterdam.

Murakami, K. and Uno, T. (2013)

Efficient Algorithms for Dualizing Large-Scale Hypergraphs.

ALENEX, New Orleans, USA, pp. 113.

Thanks!