Simplification logic for functional dependencies

Pablo Cordero (University of Malaga, Spain)

DATA ANALYSIS AND MODELING LAB
Palacky University, Olomouc, Czech Republic

european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
INVESTMENTS IN EDUCATION DEVELOPMENT

Simplification logic for functional dependencies

Pablo Cordero (University of Malaga, Spain)

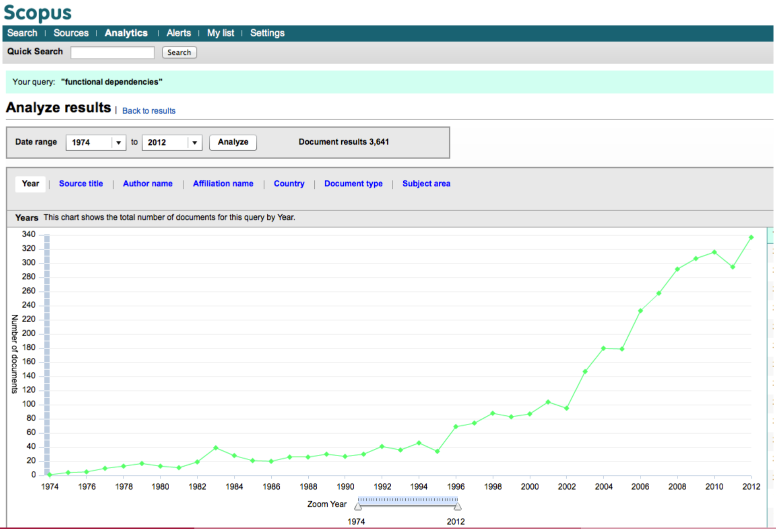
DAMOL
DATA ANALYSIS AND MODELING LAB
Palacky University, Olomouc, Czech Republic

european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
INVESTMENTS IN EDUCATION DEVELOPMENT

Outline

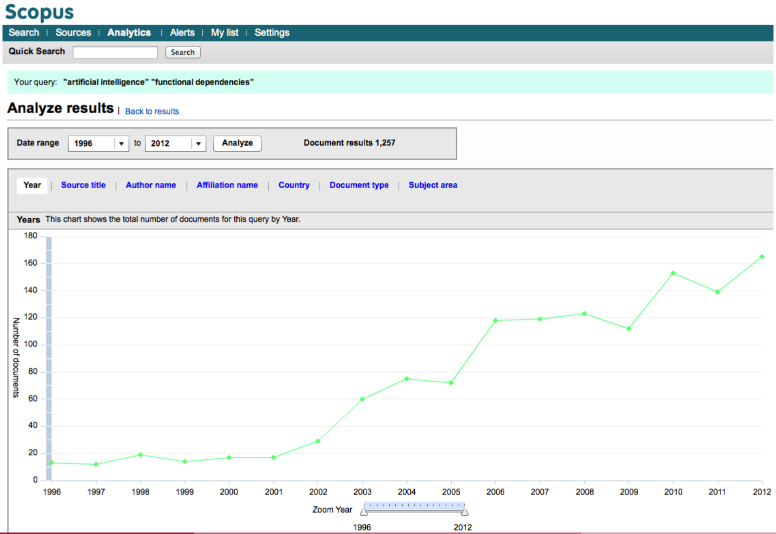
Functional Dependencies

- Relational Model [Codd, 1970]
- Functional Dependencies [Armstrong, 1974]



Functional Dependencies in Artificial Intelligence

Logic Programming, Functional Programming, Specification, Neuronal Networks, Grid Resources Management, Software Ingenieering, Formal Concepts Analysis, Description Logic, Linked Data and Web Semantic ...



Functional dependencies

- The success of the relational model is because it is easy to represent data.
- Constrains are necessary to avoid problems such as inconsistencies, redundancies, anomalies. . .

	Subject	Identity Card	Surname	Name	Course
t1	Algebra	22222222A	SMITH	RALPH	3
t2	Algebra	33333333A	ROSE	PETER	1
t3	Calculus	22222222A	SMITH	RALPH	3
t4	Calculus	44444444B	BRANDON	ANNE	4
t5	Calculus	11111111C	BUGLE	LOUISE	2
t6	Numerical Methods	33333333A	ROSE	PAUL	1

RELATIONAL DATA BASE

- Valuable functions: Computed by formulas $f(\text{Closed Call}) = \text{Registration Fee}$
- Functions given by extension: These mappings are defined in the table.

Functional dependencies and Normalitation

- $A \to B$: Values of A determine values of B.
- There exists a (partial) mapping from A to B.

Example: Database with contributions to conferences.

$Title$	$Author$	$Filiation$	$Conference$	$City$	$Year$
...	...	...	...	...	...
...	...	...	CLA	Málaga	2012
...	...	...	...	...	...
...	...	...	CLA	Nancy	2011
...	...	...	...	...	...
...	...	...	CLA	Málaga	2012
...	...	...	...	...	...

$$Title, Author \to Conference, Year, \quad Author \to Filiation, \quad Conference, Year \to City$$

$Title$	$Author$	$Conference$	$Year$
...	...	...	...
...	...	CLA	2012
...	...	...	...
...	...	CLA	2011
...	...	...	...
...	...	CLA	2012
...	...	...	...

$Author$	$Filiation$
...	...
...	...

$Conference$	$City$	$Year$
CLA	Nancy	2011
CLA	Málaga	2012
...	...	...

Functional dependencies and Normalitation

- $A \to B$: Values of A determine values of B.
- There exists a (partial) mapping from A to B.

Example: Database with contributions to conferences.

$Title$	$Author$	$Filiation$	$Conference$	$City$	$Year$
...	...	...	...	...	...
...	...	...	CLA	Málaga	2012
...	...	...	...	...	...
...	...	...	CLA	Nancy	2011
...	...	...	...	...	...
...	...	...	CLA	Málaga	2012
...	...	...	...	...	...

$$Title, Author \to Conference, Year, \quad Author \to Filiation, \quad Conference, Year \to City$$

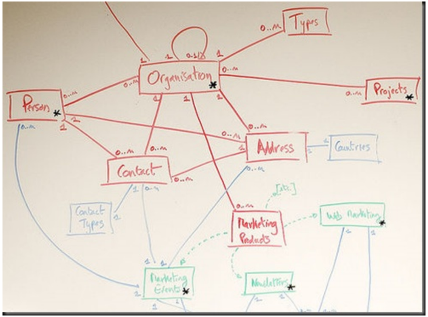
$Title$	$Author$	$Conference$	$Year$
...	...	...	...
...	...	CLA	2012
...	...	...	...
...	...	CLA	2011
...	...	...	...
...	...	CLA	2012
...	...	...	...

$Author$	$Filiation$
...	...
...	...

$Conference$	$City$	$Year$
CLA	Nancy	2011
CLA	Málaga	2012
...	...	...

Functional dependencies and Normalitation

- FDs are essential and represent the semantic of the data.
- How can I use FDs to optimize the design of the database?
- There are not efficient automatic tools for the management of FDs to ease the normalization process. Thus, FDs are in hands of gurus.



Functional dependencies and Normalization

Actual situation:

- Nowadays, the space of storage is not a problem.
- Unfortunately, today, normalization is being forgotten in the database design.
- In huge non-normalized database the consults are slower.
- The semantic of the relationship between the data is lost.
- When database degenerates, companies must repair the bad-design (over-cost).
- In some tools is possible specify FDs (Oracle) but none incorporates algorithms for FDs.
- Commercial tools not incorporate FDs because they do not know how manage it.

FDs have been left to one side!!!

Functional dependencies

Let Ω be a set of attributes, $\{D_a \mid a \in \Omega\}$ a family of domains of such attributes and $R \subseteq \prod_{a\in\Omega} D_a$ a relation.

Definition

Let $A, B \subseteq \Omega$. The relation R *satisfies* the functional dependency $A \to B$ if

$$t_A = t'_A \text{ implies } t_B = t'_B$$

for all $t, t' \in R$.

That is, $R_{A\cup B}\colon R_A \to R_B$.

$$\left.\begin{array}{rcl} \text{idCard} & \to & \text{Surname, Name} \\ \text{Surname, idCard} & \to & \text{Antiqueness, Degree} \\ \text{Antiqueness, Degree} & \to & \text{Salary} \end{array}\right\} \vdash \text{idCard} \to \text{Salary}$$

Explicit and implicit information

$$\left.\begin{array}{rcl} \text{idCard} & \to & \text{Surname, Name} \\ \text{Surname, idCard} & \to & \text{Antiqueness, Degree} \\ \text{Antiqueness, Degree} & \to & \text{Salary} \end{array}\right\} \vdash \quad \text{idCard} \to \text{Salary}$$

Armstrong's axioms

Consists on the following axioms

$\vdash \quad A \to B, \qquad$ if $B \subseteq A$ .... Axioms

and the following inference rules:

$A \to B,\ B \to C \quad \vdash \quad A \to C$ .... Transitivity

$A \to B \quad \vdash \quad A \cup C \to B \cup C$ .... Augmentation

Armstrong's axioms appear in a big number of problems in Artificial Intelligence and not always related to functional dependencies.

Looking for a more general framework

Definition (Demetrovics and Thi, 1995)

Let Ω be a finite set. A full family is a relation $\mathcal{F} \subseteq 2^U \times 2^U$ that is reflexive, transitive and satisfies de following conditions:

1. If $(A, B) \in \mathcal{F}$ and $C \subseteq B$ then $(A, C) \in \mathcal{F}$.
2. If $(A, B), (C, D) \in \mathcal{F}$ then $(A \cup C, B \cup D) \in \mathcal{F}$.

Theorem

If Ω is finite, R defines a full family:

$$F_R = \{(A, B) \mid A, B \subseteq \Omega \text{ and } R \text{ satisfies } A \to B\}$$

Conversely, for all full family $\mathcal{F}$, there exist a relation R such that $F_R = \mathcal{F}$.

Full families

Theorem (Demetrovics and Thi, 1995)

For all $A, A', B, B', C, D \subseteq \Omega$,

1. *If* $B \subseteq A$ *then* $A \to B \in F_R$.
2. *If* $A \to B \in F_R$ *then* $A \to AB \in F_R$.
3. *If* $A \to B \in F_R$ *and* $C \subseteq B$ *then* $A \to C \in F_R$.
4. *If* $A \to B, B \to C \in F_R$ *then* $A \to C \in F_R$.
5. *If* $A \to B, A \to C \in F_R$ *then* $A \to BC \in F_R$ *and* $A \to B \cap C \in F_R$.
6. *If* $A \to B, A' \to B' \in F_R$ *then* $AA' \to BB' \in F_R$.
7. *If* $A \to B \in F_R$ *then* $A \to B \smallsetminus A \in F_R$.
8. *If* $A \to B \in F_R$, $A \subseteq C$ *and* $D \subseteq AB$ *then* $C \to D \in F_R$.
9. *If* $A \to B \in F_R$ *and* $a \in B$ *then* $A \to a \in F_R$.
10. *If* $A \to B, A' \to B' \in F_R$, $A' \subseteq AB$, $A \subseteq C$ *and* $D \subseteq B'C$ *then* $C \to D \in F_R$.

Non-Deterministic Ideal Operators, *Discrete Applied Mathematics 156(6): 911-923*

- Hyperalgebras [Marty, 1934] with applications in Soft Computing.

  A Survey can be found in LNCS 6692, pp. 437–444, 2011.

- Residuated multilattices. Fuzzy Set and Systems, http://dx.doi.org/10.1016/j.fss.2013.04.002. Aslo presented in the International Workshop "Information, Uncertainty, and Imprecision" Olomouc, July 2012: http://mcin.upol.cz/WIUI-2012/page/911/

- Hyperoperation or Non-Deterministic Operation. An n-ary hyperoperation in a set U is a mapping $\star\colon U^n \to 2^U$.

  As usual, hyperoperations can be applied to sets:

$$u_1 \star \cdots \star u_{i-1} \star X \star u_{i+1} \star \cdots \star u_n = \bigcup_{x \in X} (u_1 \star \cdots \star u_{i-1} \star x \star u_{i+1} \star \cdots \star u_n)$$

- Particularly, we center on the monoid of unary non-deterministic operations $(\mathcal{N}do(U), \circ, Id_u)$

$$\mathcal{N}do(U) = \{F : U \to 2^U\} \qquad (G \circ F)(x) = \bigcup_{y \in F(x)} G(y) \qquad Id_U(x) = \{x\}$$

Non-Deterministic Ideal Operators, *Discrete Applied Mathematics 156(6): 911-923*

Definition

Let $\mathbb{L} = (L, \vee, \wedge)$ be a lattice. An unary non-deterministic operation $F \in \mathcal{N}do(L)$ is said to be an non-deterministic ideal operator if the following conditions hold:

- Reflexivity: $Id_L \subseteq F$
- Transitivity: $F \circ F \subseteq F$
- $F(a)$ is an ideal in $\mathbb{L}$, for all $a \in L$.

Theorem

For all relation R, the following mapping is a non-deterministic ideal operator in the lattice 2^Ω:

$$F_R(A) = \{B \subseteq \Omega \mid R \textit{ satisfies } A \to B\}$$

Conversely, for all non-deterministic ideal operator F in 2^Ω, there exists a relation R such that

$$F = F_R$$

Non-Deterministic Ideal Operators, *Discrete Applied Mathematics 156(6): 911-923*

Thus, we consider hyperalgebras $(L, \vee, \wedge, F)$ where

- $(L, \vee, \wedge)$ is a lattice.
- F is a non-deterministic ideal operator:

  (1) $Id_L \subseteq F$, (2) $F \circ F \subseteq F$, and (3) $F(a)$ is an ideal in $(L, \vee, \wedge)$, for all $a \in L$.

Definition

A non-deterministic ideal operator F in a lattice $\mathbb{L} = (L, \vee, \wedge)$ is said to be principal if, for all element $a \in L$, $F(a)$ is a principal ideal in $\mathbb{L}$, i.e. there exists the maximum element m in $F(a)$ such that $F(a) = (m] = \{x \in L \mid x \leq m\}$.

Non-Deterministic Ideal Operators, *Discrete Applied Mathematics 156(6): 911-923*

Theorem

Let $\mathbb{L} = (L, \vee, \wedge)$ *be a lattice.*

- *For all closure operator* c *in* $\mathbb{L}$*, the following mapping is a non-deterministic ideal operator in the lattice* $\mathbb{L}$*:*

$$F_c(a) = \{b \in L \mid b \leq c(a)\}$$

- *Conversely, for all principal non-deterministic ideal operator* F *in* $\mathbb{L}$*, the following mapping* $c\colon L \to L$ *is a closure operator in* $\mathbb{L}$

$$c(a) = \max F(a)$$

Example

The following mapping is a (non-principal) non-deterministic ideal operator in $(2^{\mathbb{R}}, \cup, \cap)$:

$$F(A) = \begin{cases} 2^{\mathbb{R}} & \text{if } A \text{ is infinite.} \\ \{X \subseteq \mathbb{R} \mid X \text{ is finite}\} & \text{if } A \text{ is finite.} \end{cases}$$

Simplification Logic. LNCS Vol. 2527, 2002, pp 141-150.

- The semantic of Simplification Logic can be defined by using non-deterministic ideal operators (in a kind of lattices) as models.
- We are going to center in the particular case of Functional Dependencies.

Language

Given a set of attributes Ω, the *language* is the set of formulas

$$\mathcal{L} = \{A \to B \mid A, B \subseteq \Omega\}$$

Following the usual notation in the database community, AB denotes $A \cup B$.

Simplification Logic. LNCS Vol. 2527, 2002, pp 141-150.

Semantic

A model for a formula $A \to B$ is a pair $(\{D_a \mid a \in \Omega\}, R)$ such that $R \subseteq \prod_{a\in\Omega} D_a$ satisfies $A \to B$.

That is, for all $t = (t_a \mid a \in \Omega) \in R$ and $t' = (t'_a \mid a \in \Omega) \in R$,

$$\text{if } t_a = t'_a \text{ for all } a \in A \text{ then } t_b = t'_b \text{ for all } b \in B$$

If no confusion arises, R denotes the model and, as usual,

- $R \models A \to B$ denotes that R is a model for $A \to B$.
- $R \models \mathrm{T}$ denotes that R is a model for all $A \to B \in \mathrm{T}$.
- $\mathrm{T} \models A \to B$ denotes that, for all model R,

$$R \models \mathrm{T} \quad \text{implies} \quad R \models A \to B$$

Implication problem: Is $\mathrm{T} \models A \to B$ true?

Simplification Logic. LNCS Vol. 2527, 2002, pp 141-150.

Axiomatic system

It has the following set of axioms:

- Reflexivity: $\vdash A \to A$

and the following inference rules:

- Decomposition: If $C \subseteq B$ then $A \to B \vdash A \to C$
- Composition: $A \to B,\ C \to D \vdash AC \to BD$
- Simplification: If $A \cap B = \varnothing$ and $A \subseteq C$ then $A \to B,\ C \to D \vdash C \smallsetminus B \to D \smallsetminus B$

being $A, B, C, D \subseteq \Omega$.

Simplification Logic. LNCS Vol. 2527, 2002, pp 141-150.

- Semantic inference: $\mathrm{T} \models A \to B$.
- Syntactic derivation: $\mathrm{T} \vdash A \to B$.

Theorem (Soundness and completeness)

For all formula $A \to B$ and all theory T,

$$\mathrm{T} \models A \to B \quad \textit{if and only if} \quad \mathrm{T} \vdash A \to B$$

- **It is not useful if we have not an automated reasoning method.**

Removing redundant information

- By removing redundant information in the specification of a database we can reduce the size of data tables and avoid problems related to inconsistencies, redundancies, anomalies...
- *Duquenne-Guigues Basis* is minimum w.r.t. the number of implications. However, redundant attributes can be appear inside of the implications.

Example [Ganter 1987, pages 30 and 84]

130 countries and six attributes are considered: Group of 77, Non-aligned, LLDC (Least Developed Countries), MASC (Most Seriously Affected Countries), OPEC (Organization of Petrol Exporting Countries) and ACP (African, Caribbean and Pacific Countries).

OPEC	$\longrightarrow$	Group 77, Non-aligned
MASC	$\longrightarrow$	Group 77
Non-aligned	$\longrightarrow$	Group 77
Group 77, Non-aligned, MASC, OPEC	$\longrightarrow$	LLDC, ACP
Group 77, Non-aligned, LLDC, OPEC	$\longrightarrow$	MASC, ACP

Removing redundant information

Definition

A theory T is said to be

- Left-redundant if, there exists $A \to B \in \mathrm{T}$ and $A' \subsetneq A$ such that
$$\mathrm{T} \equiv (T \smallsetminus \{A \to B\}) \cup \{A' \to B\}$$
- Right-redundant if, there exists $A \to B \in \mathrm{T}$ and $B' \subsetneq B$ such that
$$\mathrm{T} \equiv (T \smallsetminus \{A \to B\}) \cup \{A \to B'\}$$
- Redundant if it is left-redundant, right-redundant, or there exists $A \to B \in \mathrm{T}$ such that
$$\mathrm{T} \equiv T \smallsetminus \{A \to B\}$$

Removing redundant information

The inference rules in Simplification Logic can be translated to equivalence rules.

Theorem

Let $A, B, C, D \subseteq \Omega$.

- *Decomposition Equivalency:* $\{A \to B\} \equiv \{A \to B \smallsetminus A\}$
- *Composition Equivalency:* $\{A \to B,\ A \to C\} \equiv \{A \to BC\}$
- *Simplification Equivalency: If* $A \cap B = \varnothing$ *and* $A \subseteq C$ *then*
$$\{A \to B,\ C \to D\} \equiv \{A \to B,\ C \smallsetminus B \to D \smallsetminus B\}$$
- *r-Simplification Equivalency: If* $A \cap B = \varnothing$ *and* $A \subseteq C \cup D$ *then*
$$\{A \to B,\ C \to D\} \equiv \{A \to B,\ C \to D \smallsetminus B\}$$

Maier's algorithm

Algorithm 1: LEFTREDUCE

Data: A theory T_1
Result: A non-left-redundant equivalent theory T_2
begin
  $T_2 := T_1$;
  foreach $A \to B \in T_1$ **do**
    foreach $a \in A$ **do**
      if $T_2 \vdash A \smallsetminus \{a\} \to B$ **then** $T_2 := (T_2 \smallsetminus \{A \to B\}) \cup \{A \smallsetminus a \to B\}$
  return T_2
end

Maier's algorithm

Algorithm 2: RIGHTREDUCE

Data: A non-left-redundant equivalent theory T_1
Result: A non-left-redundant and non-right-redundant equivalent theory T_2

begin
  $T_2 := T_1$;
  foreach $A \to B \in T_1$ **do**
    foreach $b \in B$ **do**
      if $(T_2 \smallsetminus \{A \to B\}) \cup \{A \to B \smallsetminus b\} \vdash A \to b$ **then** $T_2 := (T_2 \smallsetminus \{A \to B\}) \cup \{A \to B \smallsetminus b\}$
  return T_2
end

Algorithm 3: REDUCE

Data: A theory T_1
Result: A non-redundant equivalent theory T_2

begin
  $T_2 :=$ RIGHTREDUCE(LEFTREDUCE(T_1));
  foreach $A \to B \in T_2$ **do**
    if $B = \varnothing$ **then** $T_2 := T_2 \smallsetminus \{A \to B\}$
  return T_2
end

By using Maier's algorithm:

Example

$T = \{ab \to c, c \to a, bc \to d, acd \to b, d \to eg, be \to c, cg \to bd, ce \to ag\}$

LEFTREDUCE

$T \nvdash a \to c$, $T \nvdash b \to c$, $T \nvdash \varnothing \to a$, $T \nvdash b \to d$, $T \nvdash c \to d$, $T \vdash cd \to b$, $T \nvdash ad \to b$, $T \nvdash ac \to b$, $T \nvdash \varnothing \to eg$, $T \nvdash b \to c$, $T \nvdash e \to c$, $T \nvdash c \to bd$, $T \nvdash g \to bd$, $T \nvdash c \to ag$, $T \nvdash e \to ag$.

Result

$\{ab \to c, c \to a, bc \to d, cd \to b, d \to eg, be \to c, cg \to bd, ce \to ag\}$

By using Maier's algorithm:

Example

$T = \{ab \to c, c \to a, bc \to d, acd \to b, d \to eg, be \to c, cg \to bd, ce \to ag\}$

LEFTREDUCE – Result

$\{ab \to c, c \to a, bc \to d, cd \to b, d \to eg, be \to c, cg \to bd, ce \to ag\}$

RIGHTREDUCE

$T \setminus \{ab \to c\} \nvdash ab \to \varnothing,\quad T \setminus \{c \to a\} \nvdash c \to \varnothing,\quad T \setminus \{bc \to d\} \nvdash bc \to \varnothing,$
$T \setminus \{cd \to b\} \nvdash cd \to \varnothing,\quad T \setminus \{d \to eg\} \nvdash d \to e,\quad T \setminus \{d \to eg\} \nvdash d \to g,$
$T \setminus \{be \to c\} \nvdash be \to \varnothing,\quad T \setminus \{cg \to bd\} \nvdash cg \to b,\quad T \setminus \{cg \to bd\} \vdash cg \to d,$
$T \setminus \{ce \to ag\} \vdash ce \to a,\quad T \setminus \{ce \to ag\} \nvdash ce \to a.$

Result

$\{ab \to c, c \to a, bc \to d, cd \to b, d \to eg, be \to c, cg \to b, ce \to g\}$

By using Armstrong's axioms

Example

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to eg, be \to c, cg \to bd, ce \to ag\} \equiv$ Frag

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to e, d \to g, be \to c, cg \to b, cg \to d, ce \to a, ce \to g\} \equiv$ Aug

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to e, d \to g, be \to c, cg \to b, cg \to d, ce \to g\} \equiv$ Ax

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to e, d \to g, be \to c, cg \to b, cg \to d, ce \to g\} \cup \{cg \to c\} \equiv$ Un

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to e, d \to g, be \to c, cg \to b, cg \to d, ce \to g\} \cup \{cg \to bc\} \equiv$ Tr

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to e, d \to g, be \to c, cg \to b, ce \to g\} \equiv$ Aug

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to e, d \to g, be \to c, cg \to b, ce \to g\} \cup \{c \to ac\} \equiv$ Com

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to e, d \to g, be \to c, cg \to b, ce \to g\} \cup \{cd \to acd\} \equiv$ Tr

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to e, d \to g, be \to c, cg \to b, ce \to g\} \cup \{cd \to acd, cd \to b\} \equiv$ Frag

$\{ab \to c, c \to a, bc \to d, cd \to b, d \to e, d \to g, be \to c, cg \to b, ce \to g\}$

By using Simplification Logic:

Example

$\{ab \to c, c \to a, bc \to d, acd \to b, d \to eg, be \to c, cg \to bd, ce \to ag\} \equiv$ Simp

$\{ab \to c, c \to a, bc \to d, cd \to b, d \to eg, be \to c, cg \to bd, ce \to ag\} \equiv$ Simp

$\{ab \to c, c \to a, bc \to d, cd \to b, d \to eg, be \to c, cg \to bd, ce \to g\} \equiv$ r-Simp

$\{ab \to c, c \to a, bc \to d, cd \to b, d \to eg, be \to c, cg \to b, ce \to g\}$

Removing redundant information

- In the 82% of the cases Simplification removes all redundant attributes.
- We are working in the study of new algorithms related with this aim: Optimal sets, minimum size...
- *"Unfortunately, there is probably no polynomial time algorithm for finding an optimal cover for a set of FDs. This problem belongs to the class of NP- complete problems, for which no one has yet found any polynomial time algorithms. Another NP-complete problem concerning covers is, what is the smallest set I; contained in G that is a cover for G? Size in this case is measured in FDs."* [Maier, 1980]

Implication Problem and Closures

The closure of a set of attributes A w.r.t. a set of formulas Γ, denoted as A^+, is defined as

Definition

A^+ is the greatest set of attributes such that $\Gamma \vdash A \to A^+$.

Thus, due to Fragmentation rule, for any set of attributes B,

$$\Gamma \vdash A \to B \quad \text{if and only if} \quad B \subseteq A^+$$

Theorem

$$\Gamma \vdash A \to B \ \ \textit{if and only if} \ \ \Gamma \cup \{\varnothing \to A\} \vdash \varnothing \to B$$

Implication Problem and Closures

The closure of a set of attributes A w.r.t. a set of formulas Γ, denoted as A^+, is defined as

Definition

A^+ is the greatest set of attributes such that $\Gamma \vdash A \to A^+$.

Thus, due to Fragmentation rule, for any set of attributes B,

$$\Gamma \vdash A \to B \quad \text{if and only if} \quad B \subseteq A^+$$

Theorem

$$\Gamma \vdash A \to B \;\; \textit{if and only if} \;\; \Gamma \cup \{\varnothing \to A\} \vdash \varnothing \to B$$

Implication Problem and Closures

Theorem

$$\Gamma \vdash A \to B \;\; \textit{if and only if} \;\; \Gamma \cup \{\varnothing \to A\} \vdash \varnothing \to B$$

Theorem

Let $A, B, C \subseteq Y$. The following equivalences hold:

- **S+C Eq.**: *If $B \subseteq A$ then $\{\varnothing \to A, B \to C\} \equiv \{\varnothing \to A \cup C\}$.*
- **S+A Eq.**: *If $C \subseteq A$ then $\{\varnothing \to A, B \to C\} \equiv \{\varnothing \to A\}$.*
- **S Eq.**: *Otherwise $\{\varnothing \to A, B \to C\} \equiv \{\varnothing \to A, B \smallsetminus A \to C \smallsetminus A\}$.*

Automated reasoning method to obtain the closure

From Γ and A, calculate A^+ (the closure of A):

- Add $\varnothing \to A$
- Apply systematically the three previous equivalences.

Result: $\varnothing \to A^+$

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} \quad \text{(S Eq)}$$
$$\equiv \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} \quad \text{(S+C Eq)}$$
$$\equiv \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} \quad \text{(S+C Eq)}$$
$$\equiv \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} \quad \text{(S+A Eq)}$$
$$\equiv \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} \quad \text{(S+C Eq)}$$
$$\equiv \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} \quad \text{(S+A Eq)}$$
$$\equiv \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} \quad \text{(S+C Eq)}$$
$$\equiv \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} \quad \text{(S+AEq)}$$
$$\equiv \{cf \to bh, lm \to n, \varnothing \to abcdefg\} \quad \text{(S+C Eq)}$$
$$\equiv \{lm \to n, \varnothing \to abcdefgh\}$$

The result is $bd^+ = abcdefgh$.

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} \quad \text{(S Eq)}$$

$$\equiv \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} \quad \text{(S+A Eq)}$$

$$\equiv \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} \quad \text{(S+A Eq)}$$

$$\equiv \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} \quad \text{(S+AEq)}$$

$$\equiv \{cf \to bh, lm \to n, \varnothing \to abcdefg\} \quad \text{(S+C Eq)}$$

$$\equiv \{lm \to n, \varnothing \to abcdefgh\}$$

The result is $bd^+ = abcdefgh$.

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$
\begin{array}{lll}
 & \{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S Eq)} \\
\equiv & \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+AEq)} \\
\equiv & \{cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+C Eq)} \\
\equiv & \{lm \to n, \varnothing \to abcdefgh\} &
\end{array}
$$

The result is $bd^+ = abcdefgh$.

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$
\begin{array}{llr}
 & \{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S Eq)} \\
\equiv & \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+AEq)} \\
\equiv & \{cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+C Eq)} \\
\equiv & \{lm \to n, \varnothing \to abcdefgh\} &
\end{array}
$$

The result is $bd^+ = abcdefgh$.

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} \quad \text{(S Eq)}$$

$$\equiv \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} \quad \text{(S+A Eq)}$$

$$\equiv \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} \quad \text{(S+A Eq)}$$

$$\equiv \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} \quad \text{(S+AEq)}$$

$$\equiv \{cf \to bh, lm \to n, \varnothing \to abcdefg\} \quad \text{(S+C Eq)}$$

$$\equiv \{lm \to n, \varnothing \to abcdefgh\}$$

The result is $bd^+ = abcdefgh$.

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$\begin{array}{llr}
 & \{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S Eq)} \\
\equiv & \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+AEq)} \\
\equiv & \{cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+C Eq)} \\
\equiv & \{lm \to n, \varnothing \to abcdefgh\} &
\end{array}$$

The result is $bd^+ = abcdefgh$.

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$
\begin{array}{lll}
& \{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S Eq)} \\
\equiv & \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+AEq)} \\
\equiv & \{cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+C Eq)} \\
\equiv & \{lm \to n, \varnothing \to abcdefgh\} &
\end{array}
$$

The result is $bd^+ = abcdefgh$.

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$
\begin{array}{llr}
 & \{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S Eq)} \\
\equiv & \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+AEq)} \\
\equiv & \{cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+C Eq)} \\
\equiv & \{lm \to n, \varnothing \to abcdefgh\} &
\end{array}
$$

The result is $bd^+ = abcdefgh$.

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$\begin{array}{llr}
 & \{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S Eq)} \\
\equiv & \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+A Eq)} \\
\equiv & \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} & \text{(S+C Eq)} \\
\equiv & \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+AEq)} \\
\equiv & \{cf \to bh, lm \to n, \varnothing \to abcdefg\} & \text{(S+C Eq)} \\
\equiv & \{lm \to n, \varnothing \to abcdefgh\} &
\end{array}$$

The result is $bd^+ = abcdefgh$.

Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} \quad \text{(S Eq)}$$

$$\equiv \{a \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bde\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} \quad \text{(S+A Eq)}$$

$$\equiv \{a \to c, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bcdeg\} \quad \text{(S+C Eq)}$$

$$\equiv \{a \to c, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} \quad \text{(S+A Eq)}$$

$$\equiv \{a \to c, cf \to bh, cg \to af, lm \to n, \varnothing \to abcdeg\} \quad \text{(S+C Eq)}$$

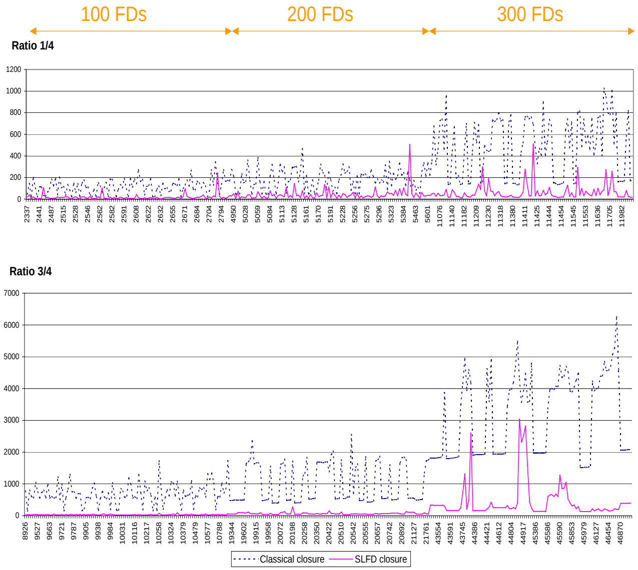
$$\equiv \{a \to c, cf \to bh, lm \to n, \varnothing \to abcdefg\} \quad \text{(S+AEq)}$$

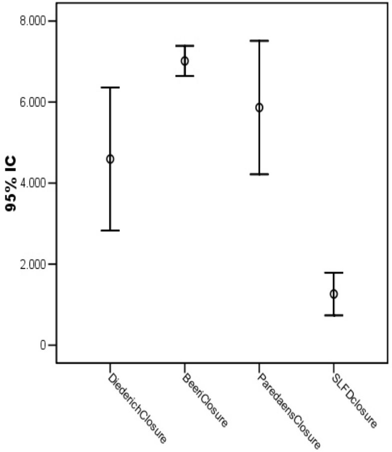
$$\equiv \{cf \to bh, lm \to n, \varnothing \to abcdefg\} \quad \text{(S+C Eq)}$$

$$\equiv \{lm \to n, \varnothing \to abcdefgh\}$$

The result is $bd^+ = abcdefgh$.

Simplification Logic vs Closure Algorithms





Example

Let us compute the closure of bd from

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n\}$$

$$\{ad \to c, b \to e, be \to cg, bc \to g, c \to a, cd \to b, cf \to bh, cg \to af, lm \to n, \varnothing \to bd\}$$

$$\equiv \{ \qquad lm \to n, \varnothing \to abcdefgh\}$$

The result is $bd^+ = abcdefgh$.

- Cordero, Enciso, Mora, Ojeda-Aciego. *Computing minimal generators from implications: a logic-guided approach.* CLA 2012. http://cla.inf.upol.cz/papers/cla2012/paper16.pdf

Fuzzy Extensions

- Functional Dependencies over domains with similarities.

  Cordero, Enciso, Mora, Pérez de Guzmán. *A complete logic for fuzzy functional dependencies over domains with similarity relations*. LNCS 5517: 261–269. 2009.

- Graded Functional Dependencies over domains with similarities.

  Cordero, Enciso, Mora, Pérez de Guzmán, Rodríguez-Jiménez. *An efficient algorithm for reasoning about fuzzy functional dependencies.* LNCS 6692: 412–420. 2011.

- FASL: Fuzzy Attribute Simplification Logic.

  Bělohlávek, Cordero, Enciso, Mora, Vychodil. *An Efficient Reasoning Method for Dependencies over Similarity and Ordinal Data*. LNCS 7647: 408–419. 2012

Outline

Simplification logic for functional dependencies

Pablo Cordero (University of Malaga, Spain)

DATA ANALYSIS AND MODELING LAB

Palacky University, Olomouc, Czech Republic

european social fund in the czech republic

EUROPEAN UNION

MINISTRY OF EDUCATION, YOUTH AND SPORTS

OP Education for Competitiveness

INVESTMENTS IN EDUCATION DEVELOPMENT