Dependencies with FCA and Pattern Structures: a Tutorial

Jaume Baixeries

Departament de Ciències de la Computació

Universitat Politècnica de Catalunya, 08034 Barcelona, Catalonia

jbaixer@cs.upc.edu

ICFCA Tutorial. June 29 2021

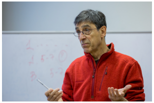
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament de Ciències de la Computació

About this Tutorial

- This tutorial show the most relevant results that we have been presenting in FCA venues and journals during the last years.
- We will present our results by the example. You always can refer to our papers for technical details.

Authors

This is a joint work that started in 2012 between:



Amedeo Napoli. Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France.

Mehdi Kaytoue. Infologic. 19 rue Victorien Sardou – 69007 LYON. France.

Víctor Codocedo. Universidad Técnica Federico Santa María. Departamento de Informática. San Joaquín, Santiago, Chile.

Jaume Baixeries. Universitat Politècnica de Catalunya. Departament de Ciències de la Computació. Barcelona. Catalonia.

Summary of the presentation

What is a Dependency?

- Functional dependencies, multivalued dependencies, join dependencies, etc, are defined in all texts.
- In Database books, a dependency (without any other modifier) is not defined.

However, the concept of dependency is familiar to everyone.

What is a Dependency?

Dependencies, in general, are semantically meaningful and syntactically restricted sentences of the predicate calculus that must be satisfied by any "legal" database.

Paris C. Kanellakis

In *Handbook of Theoretical Computer Science. Volume B Formal Models and Semantics by Jan van Leeuwen (ed.)*

What is a Dependency?

Definition + Usage

1. semantically meaningful
2. syntactically restricted sentences of the predicate calculus
3. that must be satisfied by any "legal" database.

What is a Dependency?

Definition

1. semantically meaningful
2. syntactically restricted sentences of the predicate calculus
3. that must be satisfied by any "legal" database

$$\textit{dependency} = \textit{syntax}\ (\text{predicate calculus})\ + \textit{semantics}$$

What is a Dependency?

Usage

1. semantically meaningful
2. syntactically restricted sentences of the predicate calculus
3. that must be satisfied by any "legal" database

$$dependency = restriction$$

What is a Dependency?

The usage of dependencies can be seen from two points of view

1. As a restriction that must apply to a dataset. This is a a priori point of view: it must apply before we create the dataset. This is the point of view of the database practitioners.
2. As a description of a pattern that exist in a dataset. This is the point of view of a data analyst.

What is a Dependency?

Dependencies may appear in different fields

- (Functional, multivalued, join, ...) dependencies in the relational database field (restriction).
- Implications, association rules, in the Knowledge Discovery field (description).

What do we do?

Instead of First Order Logic, we choose Formal Concept Analysis

We embed the semantics of a dependency into the incidence relation in FCA

What do we do?

Given a dataset

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

What do we do?

Given a dataset, we want to compute/characterize the set of dependencies that hold on it.

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

$$\Downarrow$$

$$\{Month, City\} \rightarrow \{Av.Temp\}$$
$$\ldots$$
$$\{Av.Temp\} \rightarrow \{City\}$$

What do we do?

How? We compute a formal context (or a pattern structure)

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

$$\Downarrow$$

$$\{Month, City\} \rightarrow \{Av.Temp\}$$
$$\ldots$$
$$\{Av.Temp\} \rightarrow \{City\}$$

$\mathbb{K}$	Month	Year	Av. Temp.	City
(1,2)	×	×		
(1,3)				×
(1,4)		×		
(2,3)				
(2,4)		×	×	
(3,4)	×			

What do we do?

How? We compute a formal context (or a pattern structure) such that the INTERPRETATION of a dependency in the formal context decides if this dependency holds in the dataset.

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

$\mathbb{K}$	Month	Year	Av. Temp.	City
(1,2)	×	×		
(1,3)				×
(1,4)		×		
(2,3)				
(2,4)		×	×	
(3,4)	×			

$\Downarrow$ $\qquad\qquad$ $\Downarrow$

$\{Month, City\} \rightarrow \{Av.Temp\}$

$\ldots$

$\{Av.Temp\} \rightarrow \{City\}$

$\equiv$

INTERPRETATION

What do we do?

What is the INTERPRETATION of a dependency?

We validate the dependency in the formal context (pattern structure)

Why do we do it?

- Theoretical interest. We have a unified framework to characterize different kinds of dependencies.
- Practical interest (ongoing work). We want to use FCA algorithms to compute basis of sets of dependencies.

Notation

A **table dataset** consists of a set of attributes and a set of tuples

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

$$Attributes = \mathcal{U} = \{Month, Year, Av.Temp., City\}$$

$$Tuples = T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$$

Notation

A **table dataset** consists of a set of attributes and a set of tuples

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

$$Attributes = \mathcal{U} = \{Month, Year, Av.Temp., City\}$$

$$Tuples = T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$$

Notation

We also use the restriction of a tuple

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

$t_3(< Month, City >) =< \mathbf{5}, Rome >$

Pattern Structures

- Bernhard Ganter and Sergei O. Kuznetsov. Pattern Structures and Their Projections, in Proceedings of the 9th International Conference on Conceptual Structures (ICCS-2001), LNCS 2120, pages 129–142, 2001.
- Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli and Sébastien Duplessis. Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis, Information Science, 181(10):1989–2001, 2011.
- Mehdi Kaytoue, Victor Codocedo, Aleksey Buzmakov, Jaume Baixeries, Sergei O. Kuznetsov and Amedeo Napoli. Pattern Structures and Concept Lattices for Data Mining and Knowledge Processing, in Proceedings of ECML-PKDD (European Conference on Machine Learning and Knowledge Discovery in Databases), Springer Lecture Notes in Computer Science 9286, pages 227–231, 2015.

Pattern Structure: Notation

A pattern structure $(G, (D, \sqcap), \delta)$ is composed of:

- G a set of *objects*,
- $(D, \sqcap)$ a semi-lattice of descriptions or patterns,
- $\delta : G \mapsto D$ a mapping such as $\delta(g) \in D$ describes object g.

The Galois connection for $(G, (D, \sqcap), \delta)$ is defined as:

- The maximal description representing the similarity of a set of objects:

$$A^{\square} = \sqcap_{g \in A} \delta(g) \qquad \text{for } A \subseteq G$$

- The maximal set of objects sharing a given description:

$$d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D, \sqcap)$$

Equivalence FCA and Pattern Structures

FCA $\Rightarrow$ Pattern Structures

Considering a standard formal context (G, M, I):

- G is the set of *objects*,
- $(D, \sqcap)$ corresponds to $(\wp(M), \cap)$ where M is the set of attributes.
- $\delta(g) = g'$.

Pattern Structures $\Rightarrow$ FCA

Considering a Pattern Structure $(G, (D, \sqcap), \delta)$ (representation context):

- G is the set of *objects*,
- $M \subseteq D$.
- $gIm \iff m \sqsubseteq \delta(g)$.

Representation of Binary Relations

We deal with equivalence and tolerance (or dependency) relations.

Equivalence Relation	Tolerance Relation
Equality ($=$)	Similarity ($\approx$)
Reflexivity ✓ Symmetry ✓ Transitivity ✓	Reflexivity ✓ Symmetry ✓ Transitivity ✗
Equivalence Classes	Blocks of Tolerance

Representation of Binary Relations

We have two ways to represent a binary relation $R \subseteq S \times S$

1. As a set of pairs.
2. As an enumeration of the classes/blocks of that relation.

A class or block Q is a maximal subset of S such that

$$\forall p, q \in Q : (p, q) \in R$$

Representation of Binary Relations

Example of a tolerance relation

$$(t_1, t_3), (t_2, t_3), (t_2, t_4), (t_3, t_4),$$
$$(t_3, t_1), (t_3, t_2), (t_4, t_3), (t_4, t_3),$$
$$(t_1, t_1), (t_2, t_2), (t_3, t_3), (t_4, t_4)$$

$$\equiv$$

$$\{\{t_1, t_3\}, \{t_2, t_3, t_4\}\}$$

Representation of Binary Relations

Example of an equivalence relation

$$(t_1, t_3), (t_2, t_4),$$
$$(t_3, t_1), (t_4, t_2),$$
$$(t_1, t_1), (t_2, t_2), (t_3, t_3), (t_4, t_4)$$

$$\equiv$$

$$\{\{t_1, t_3\}, \{t_2, t_4\}\}$$

Representation of Binary Relations

Since both equivalence and tolerance relations are reflexive and symmetric, we usually drop the pairs that show reflexivity and symmetry

$$\begin{array}{ll} (t_1, t_3), (t_2, t_3), (t_2, t_4), (t_3, t_4), & \\ (t_3, t_1), (t_3, t_2), (t_4, t_3), (t_4, t_3), & \text{symmetry} \\ (t_1, t_1), (t_2, t_2), (t_3, t_3), (t_4, t_4) & \text{reflexivity} \end{array}$$

$$\equiv$$

$$\{(t_1, t_3), (t_2, t_3), (t_2, t_4), (t_3, t_4)\} \quad \equiv \quad \{\{t_1, t_3\}, \{t_2, t_3, t_4\}\}$$

Representation of Binary Relations

Since both equivalence and tolerance relations are reflexive and symmetric, we usually drop the pairs that show reflexivity and symmetry

$$\begin{array}{ll} (t_1, t_3), (t_2, t_4), & \\ (t_3, t_1), (t_4, t_2), & \text{symmetry} \\ (t_1, t_1), (t_2, t_2), (t_3, t_3), (t_4, t_4) & \text{reflexivity} \end{array}$$

$$\equiv$$

$$\{(t_1, t_3), (t_2, t_4)\} \quad \equiv \quad \{\{t_1, t_3\}, \{t_2, t_4\}\}$$

Generalizing Equivalence Relations

Since both equivalence and tolerance relations can be expressed as sets of pairs of elements, the meet and join of two equivalence or tolerance relations are defined as the intersection and the union of sets (of pairs)

$$\{\{t_1, t_3\}, \{t_2, t_3, t_4\}\} \wedge \{\{t_1, t_2, t_4\}, \{t_1, t_3, t_4\}\}$$
$$\equiv$$
$$\{(t_1, t_3), (t_2, t_3), (t_2, t_4), (t_3, t_4)\} \cap \{(t_1, t_2), (t_1, t_4), (t_2, t_4), (t_1, t_3), (t_3, t_4)\}$$
$$\equiv$$
$$\{(t_1, t_3), (t_2, t_4), (t_3, t_4)\}$$
$$\equiv$$
$$\{\{t_1, t_3\}, \{t_2, t_4\}, \{t_3, t_4\}\}$$

Functional Dependencies: Definition

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

A functional dependency (FD) $X \rightarrow Y$ holds in T if

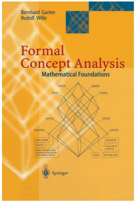
$$\forall t_i, t_j \in T : t_i(X) = t_j(X) \Rightarrow t_i(Y) = t_j(Y)$$

$a \rightarrow d$ and $d \rightarrow a$ hold
$a \rightarrow c$ does not hold.

Functional Dependencies and FCA: An Example

Functional Dependencies

- Bernhard Ganter and Rudolf Wille. Formal Concept Analysis. Mathematical Foundations. Springer.



Functional Dependencies and FCA: An Example

We want to compute the functional dependencies that hold in this table:

id	a	b	c	d
t_1	3	1	2	1
t_2	1	3	1	2
t_3	3	2	1	1
t_4	2	3	1	2

Functional Dependencies and FCA: An Example

Spoiler Alert!! These are:

$a \to d$ $\quad b \to cd$ $\quad bc \to d$ $\quad bd \to c$ $\quad ab \to cd$

$ac \to bd$ $\quad abc \to d$ $\quad abd \to c$ $\quad acd \to b$

id	a	b	c	d
t_1	3	1	2	1
t_2	1	3	1	2
t_3	3	2	1	1
t_4	2	3	1	2

(we ommit trivial FD's: $X \to Y$, where $Y \subseteq X$)

Functional Dependencies and FCA: An Example

We construct the Formal Context $\mathbb{K} = (\mathcal{B}_2(G), M, I)$

$$\mathcal{B}_2(G) = \{(t_i, t_j) \mid i < j \text{ and } t_i, t_j \in T\}$$

$$(t_i, t_j) \; I \; m \Leftrightarrow t_i(m) = t_j(m)$$

id	a	b	c	d
t_1	3	1	2	1
t_2	1	3	1	2
t_3	3	2	1	1
t_4	2	3	1	2

$\mathbb{K}$	a	b	c	d
(t_1, t_2)				
(t_1, t_3)	×			×
(t_1, t_4)				
(t_2, t_3)			×	
(t_2, t_4)		×	×	×
(t_3, t_4)			×	

Functional Dependencies and FCA: An Example

We construct the Formal Context $\mathbb{K} = (\mathcal{B}_2(G), M, I)$

$$\mathcal{B}_2(G) = \{(t_i, t_j) \mid i < j \text{ and } t_i, t_j \in T\}$$

$$(t_i, t_j)\ I\ m \Leftrightarrow t_i(m) = t_j(m)$$

id	a	b	c	d
t_1	3	1	2	1
t_2	1	3	1	2
t_3	3	2	1	1
t_4	2	3	1	2

$\mathbb{K}$	a	b	c	d
(t_1, t_2)				
(t_1, t_3)	×			×
(t_1, t_4)				
(t_2, t_3)			×	
(t_2, t_4)		×	×	×
(t_3, t_4)			×	

Functional Dependencies and FCA: An Example

We construct the Formal Context $\mathbb{K} = (\mathcal{B}_2(G), M, I)$

$$\mathcal{B}_2(G) = \{(t_i, t_j) \mid i < j \text{ and } t_i, t_j \in T\}$$

$$(t_i, t_j)\ I\ m \Leftrightarrow t_i(m) = t_j(m)$$

id	a	b	c	d
t_1	3	1	2	1
t_2	1	3	1	2
t_3	3	2	1	1
t_4	2	3	1	2

$\mathbb{K}$	a	b	c	d
(t_1, t_2)				
(t_1, t_3)	×			×
(t_1, t_4)				
(t_2, t_3)			×	
(t_2, t_4)		×	×	×
(t_3, t_4)			×	

Functional Dependencies and FCA: An Example

We have the following Formal Concepts:

$(2(G), \emptyset)$

$(\{(t_2, t_3), (t_2, t_4), (t_3, t_4)\}, \{c\})$

$(\{(t_1, t_3), (t_2, t_4)\}, \{d\})$

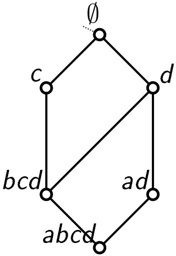
$(\{(t_1, t_3)\}, \{a, d\})$

$(\{(t_2, t_4)\}, \{b, c, d\})$

$(\emptyset, \{a, b, c, d\})$

Functional Dependencies and FCA: An Example

We draw the Concept Lattice.



$\mathbb{K}$	a	b	c	d
(t_1, t_2)				
(t_1, t_3)	×			×
(t_1, t_4)				
(t_2, t_3)			×	
(t_2, t_4)		×	×	×
(t_3, t_4)			×	

Functional Dependencies and FCA: An Example

Disclaimer!!

1. In our papers we change the orientation of the lattice (the top concept will be $(\emptyset, \{a, b, c, d\})$ and the bottom concept will be $(2(G), \emptyset)$).
2. We also remove the extents from the formal concepts.

Functional Dependencies and FCA: An Example

We can now interpret a functional dependency. A functional dependency $X \to Y$ holds in T if and only if

$$X' = XY'$$

in the formal context $\mathbb{K} = (\mathcal{B}_2(T), \mathcal{U}, I)$

$$a \to d$$

holds because

$$a' = ad'$$

$\mathbb{K}$	a	b	c	d
(t_1, t_2)				
(t_1, t_3)	$\times$			$\times$
(t_1, t_4)				
(t_2, t_3)			$\times$	
(t_2, t_4)		$\times$	$\times$	$\times$
(t_3, t_4)			$\times$	

Functional Dependencies and FCA: An Example

We can now interpret a functional dependency. A functional dependency $X \rightarrow Y$ holds in T if and only if

$$X' = XY'$$

in the formal context $\mathbb{K} = (\mathcal{B}_2(G), M, I)$

$c \rightarrow bd$

does not hold because

$c' \neq bcd'$

$\mathbb{K}$	a	b	c	d
(t_1, t_2)				
(t_1, t_3)	×			×
(t_1, t_4)				
(t_2, t_3)			×	
(t_2, t_4)		×	×	×
(t_3, t_4)			×	

Functional Dependencies as Partitions

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

The partition of T induced by $X \subseteq \mathcal{U}$ is an equivalence relation of the set of tuples T

$$\Pi_X(T) = \{c_1, c_2, \ldots, c_m\}$$

Functional Dependencies as Partitions

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

$$\Pi_a(T) = \{\{t_1, t_3\}, \{t_2, t_4\}\}$$

Functional Dependencies as Partitions

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

$$\Pi_{bc}(T) = \{\{t_1, t_2\}, \{t_3\}, \{t_4\}\}$$

Functional Dependencies as Partitions

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

A functional dependency $X \rightarrow Y$ holds in a table T if and only if

$$\Pi_X(T) = \Pi_{XY}(T)$$

$$\Pi_a(T) = \{\{t_1, t_3\}, \{t_2, t_4\}\} = \{\{t_1, t_3\}, \{t_2, t_4\}\} = \Pi_{ad}(T)$$

$$\Rightarrow$$

$a \rightarrow d$ holds

Functional Dependencies

- Jaume Baixeries, Mehdi Kaytoue and Amedeo Napoli. Characterizing Functional Dependencies in Formal Concept Analysis with Pattern Structures, Annals of Mathematics and Artificial Intelligence, 72:129–149, 2014.

FD's and Pattern Structures: An Example

We want to compute the functional dependencies that hold in this table:

id	a	b	c	d
t_1	3	1	2	1
t_2	1	3	1	2
t_3	3	2	1	1
t_4	2	3	1	2

using Pattern Structures

FD's and Pattern Structures: An Example

These are:

$a \to d$	$b \to cd$	$bc \to d$	$bd \to c$	$ab \to cd$
$ac \to bd$	$abc \to d$	$abd \to c$	$acd \to b$	

id	a	b	c	d
t_1	3	1	2	1
t_2	1	3	1	2
t_3	3	2	1	1
t_4	2	3	1	2

(we ommit trivial FD's: $X \to Y$, where $Y \subseteq X$)

FD's and Pattern Structures: An Example

We construct the Pattern Structure:

$$(M, (D, \sqcap), \delta)$$

- M is the set of attributes $\mathcal{U}$ of the original dataset.
- D is the lattice of partitions of the original table T.
- $\sqcap$ is the meet of partitions.
- $\delta(X) : \mathcal{U} \mapsto D$ is $\Pi_X(T)$.

FD's and Pattern Structures: An Example

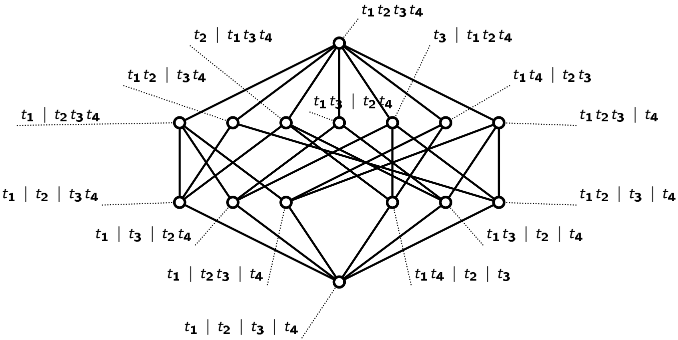
We construct the Pattern Structure $(M, (D, \sqcap), \delta)$
M is the set of attributes $\mathcal{U}$ of the original table:

$$M = \{a, b, c, d\}$$

id	a	b	c	d
t_1	3	1	2	1
t_2	1	3	1	2
t_3	3	2	1	1
t_4	2	3	1	2

FD's and Pattern Structures: An Example

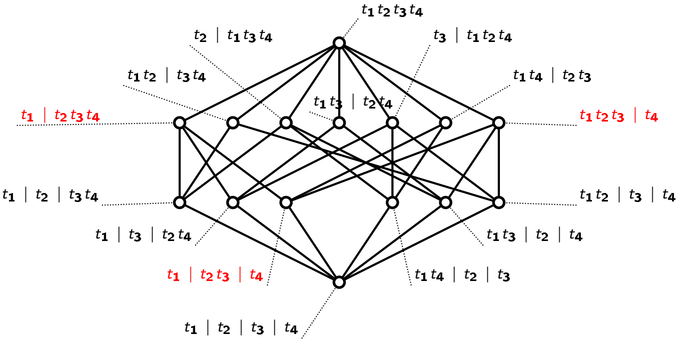
We construct the Pattern Structure $(M, (D, \sqcap), \delta)$
D is the lattice of partitions of the original table



FD's and Pattern Structures: An Example

We construct the Pattern Structure $(M, (D, \sqcap), \delta)$

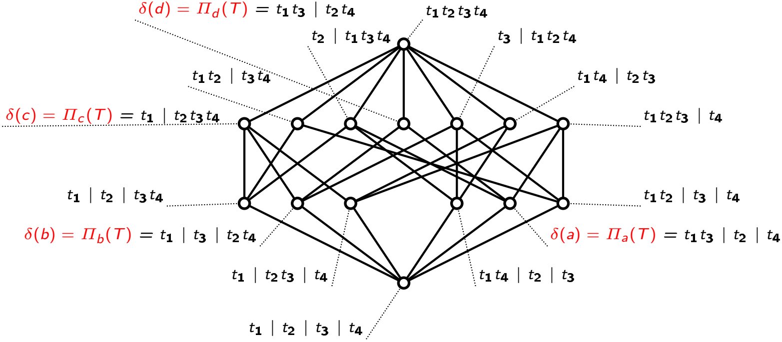
$\sqcap$ is the meet in the lattice of partitions of the original table



$$\{t_1 \mid t_2 t_3 t_4\} \sqcap \{t_1 t_2 t_3 \mid t_4\} = \{t_1 \mid t_2 t_3 \mid t_4\}$$

FD's and Pattern Structures: An Example

We construct the Pattern Structure $(M, (D, \sqcap), \delta)$
$\delta(X)$ is the function $\Pi_X(T)$



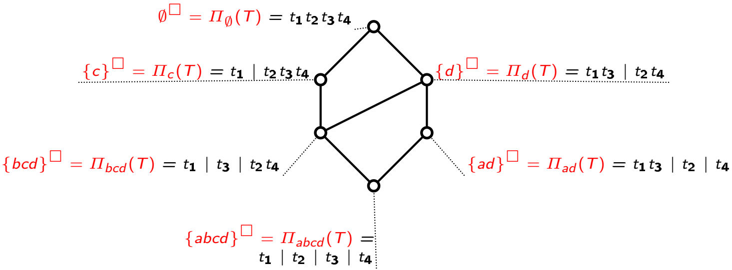
FD's and Pattern Structures: An Example

We compute the Pattern Concepts

Closed sets of Objects			Closed sets of descriptions
$\emptyset^{\square}$	$\Pi(T)_{\emptyset}$	$=$	$t_1 t_2 t_3 t_4$
$\{bcd\}^{\square}$	$\Pi(T)_{bcd}$	$=$	$t_1 \mid t_3 \mid t_2 t_4$
$\{ad\}^{\square}$	$\Pi(T)_{ad}$	$=$	$t_1 t_3 \mid t_2 \mid t_4$
$\{c\}^{\square}$	$\Pi(T)_{c}$	$=$	$t_1 \mid t_2 t_3 t_4$
$\{d\}^{\square}$	$\Pi(T)_{d}$	$=$	$t_1 t_3 \mid t_2 t_4$
$\{abcd\}^{\square}$	$\Pi(T)_{abcd}$	$=$	$t_1 \mid t_2 \mid t_3 \mid t_4$

FD's and Pattern Structures: An Example

We construct the Pattern Lattice



FD's and Pattern Structures: An Example

The interpretation of FD's in a pattern structure

A functional dependency $X \rightarrow Y$ holds in a data table T if and only if

$$\{X\}^{\square} = \{X, Y\}^{\square}$$

in the partition pattern structure $(\mathcal{U}, (Part(T), \sqcap), \Pi_X(T))$

$a \rightarrow d$ holds because

$$\Pi(T)_a = \{a\}^{\square} = \{ad\}^{\square} = \Pi(T)_{ad} = t_1 t_3 \mid t_2 \mid t_4$$

FD's and Pattern Structures: An Example

The interpretation of FD's in a pattern structure

A functional dependency $X \to Y$ holds in a data table T if and only if

$$\{X\}^\square = \{X, Y\}^\square$$

in the partition pattern structure $(\mathcal{U}, (Part(T), \sqcap), \Pi_X(T))$

REMEMBER

A functional dependency $X \to Y$ holds in a data table T if and only if

$$X' = XY'$$

in the formal context $\mathbb{K} = (\mathcal{B}_2(T), \mathcal{U}, I)$

FD's and Pattern Structures: An Example

The interpretation of FD's in a pattern structure

A functional dependency $X \rightarrow Y$ holds in a data table T if and only if

$$\{X\}^{\square} = \{X, Y\}^{\square}$$

in the partition pattern structure $(\mathcal{U}, (Part(T), \sqcap), \Pi_X(T))$

$a \rightarrow d$ holds because

$$\Pi(T)_a = \{a\}^{\square} = \{ad\}^{\square} = \Pi(T)_{ad} = t_1 t_3 \mid t_2 \mid t_4$$

FD's and Pattern Structures: An Example

A functional dependency $X \rightarrow Y$ holds in a data table T if and only if

$$\{X\}^{\Box} = \{X, Y\}^{\Box}$$

in the partition pattern structure $(\mathcal{U}, (Part(T), \sqcap), \Pi_X(T))$

REMEMBER!

Since $\{X\}^{\Box} = \delta(X) = \Pi_X(T)$

A functional dependency $X \rightarrow Y$ holds in a table T if and only if

$$\Pi_X(T) = \Pi_{XY}(T)$$

FD's and Pattern Structures: An Example

The interpretation of FD's in a pattern structure

A functional dependency $X \rightarrow Y$ holds in a data table T if and only if

$$\{X\}^{\square} = \{X, Y\}^{\square}$$

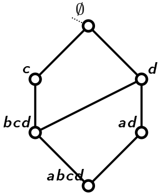
in the partition pattern structure $(\mathcal{U}, (Part(T), \sqcap), \Pi_X(T))$

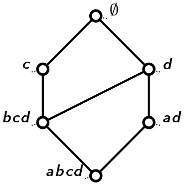
$d \rightarrow a$ does not hold because

$$t_1 t_3 \mid t_2 \mid t_4 = \{ad\}^{\square} \neq \{d\}^{\square} = t_1 t_3 \mid t_2 t_4$$

Relationship between Binarization and Pattern Structures

What is the relationship between the formal context $\mathbb{K} = (\mathcal{B}_2(G), M, I)$ and the pattern structure $(M, (D, \sqcap), \delta)$?





Both lattices are isomorphic.

In the pattern lattice, the attributes (of the table T) are the objects, whereas in the concept lattices, they are the attributes.

Relationship between Binarization and Pattern Structures

What is the relationship between the formal context $\mathbb{K} = (\mathcal{B}_2(G), M, I)$ and the pattern structure $(M, (D, \sqcap), \delta)$?

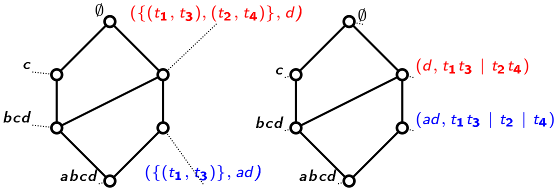
(B, A) is a pattern concept		$(M, (D, \sqcap), \delta)$
	$\Leftrightarrow$	
(A, B) is a formal concept		$(\mathcal{B}_2(G), M, I)$

Relationship between Binarization and Pattern Structures

(B, A) is a pattern concept $\qquad (M, (D, \sqcap), \delta)$

$\Leftrightarrow$

(A, B) is a formal concept $\qquad (\mathcal{B}_2(G), M, I)$



Relationship between Binarization and Pattern Structures

$$\begin{array}{lcl} (B, A) \text{ is a pattern concept} & & (M, (D, \sqcap), \delta) \\ & \Leftrightarrow & \\ (A, B) \text{ is a formal concept} & & (\mathcal{B}_2(G), M, I) \end{array}$$

The information contained is the same

$$(\{(t_1, t_3), (t_2, t_4)\}, d) \quad \equiv \quad (d, t_1 t_3 \mid t_2 t_4)$$

$$(\{(t_1, t_3)\}, ad) \quad \equiv \quad (ad, t_1 t_3 \mid t_2 \mid t_4)$$

Functional Dependencies are not enough

Slight differences in value prevent some intuitive FD's from holding

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

$$Month, City \rightarrow Av.Temp.$$

Functional Dependencies are not enough

Slight differences in value prevent some intuitive FD's from holding

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

Removing some tuples allows a dependency to exist.

For example, the dependency $Month, City \rightarrow Av.Temp$ holds if 4 tuples are removed.

Functional Dependencies are not enough

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

The idea is to have a dependency that says:

Given cities that are close, in similar months, we can determine within some interval the temperature

Functional Dependencies are not enough

id	Month	Year	Av. Temp.	City
t_1	1	1995	36.4	Milan
t_2	1	1996	33.8	Milan
t_3	5	1996	63.1	Rome
t_4	5	1997	59.6	Rome
t_5	1	1998	41.4	Dallas
t_6	1	1999	46.8	Dallas
t_7	5	1996	84.5	Houston
t_8	5	1998	80.2	Houston

We soften the definition of FDs:

$$\forall t_i, t_j \in T : t_i(X) = t_j(X) \Rightarrow t_i(Y) = t_j(Y)$$
$$\Downarrow$$
$$\forall t_i, t_j \in T : t_i(X) \approx t_j(X) \Rightarrow t_i(Y) \approx t_j(Y)$$

(where $\approx$ is user defined)

Generalizing Equivalence Relations

We switch from equivalence relations to tolerance/dependency relations

Equivalence Relation	Tolerance Relation
Equality ($=$)	Similarity ($\approx$)
Reflexivity ✓ Symmetry ✓ Transitivity ✓	Reflexivity ✓ Symmetry ✓ Transitivity ✗
Equivalence Classes	Blocks of Tolerance

Generalizing Equivalence Relations

Instead of the operator

$$\Pi_X(T)$$

that computed the partition of T induced by the set of attributes X, we define the operator

$$T/\theta_X$$

that computes the tolerance relation induced by the set of attributes X

Tolerance Relations and Blocks of Tolerance

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

We define this tolerance relation

$$t_i \theta_m t_j \iff |t_i(m) - t_j(m)| \leq 1$$

- $T/\theta_a = \{\{t_1, t_3\}, \{t_2, t_4\}\}$
- $T/\theta_b = \{\{t_1, t_2, t_4\}, \{t_3\}\}$
- $T/\theta_c = \{\{t_1, t_2, t_3\}, \{t_4\}\}$
- $T/\theta_d = \{\{t_1, t_3\}, \{t_2, t_4\}\}$

Tolerance Relations and Blocks of Tolerance

Although this definition:

$$t_i \theta_m t_j \iff \textit{their values are somehow related}$$

is very common when defining soft functional dependencies, it is not the only way to define a tolerance relation

Similarity Dependencies

- Jaume Baixeries, Victor Codocedo, Mehdi Kaytoue and Amedeo Napoli. Characterizing Approximate-Matching Dependencies in Formal Concept Analysis with Pattern Structures, Discrete Applied Mathematics, 249:18–27, 2018.

Similarity Dependencies: a Definition

Given a similarity relation θ_x (reflexive and symmetric) for each attribute x

The similarity dependency $X \rightarrow Y$ holds in a dataset T iff

$$\forall t_i, t_j \in T: \quad t_i \theta_X t_j \quad \Rightarrow \quad t_i \theta_Y t_j$$

$$\forall t_i, t_j \in T: \quad t_i(X) \approx t_j(X) \quad \Rightarrow \quad t_i(Y) \approx t_j(Y)$$

Similarity Dependencies and Partition Structures: an Example

Given the tolerance relation: $t_i \theta_m t_j \iff |t_i(m) - t_j(m)| \leq 2$ we want to compute all the similarity dependencies that hold in

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

These are:

$$a \rightarrow d, ab \rightarrow d, abc \rightarrow d, ac \rightarrow d, b \rightarrow d, bc \rightarrow d, c \rightarrow d$$

Similarity Dependencies and Partition Structures: an Example

We construct the Pattern Structure:

$$(M, (D, \sqcap), \delta)$$

- M is the set of attributes $\mathcal{U}$ of the original table.
- D is the lattice of tolerance relations of the original table.
- $\sqcap$ is the meet (intersection) of tolerance relations.
- $\delta(m) = G/\theta_m$: the tolerance relation induced by θ_m.

Similarity Dependencies and Partition Structures: an Example

The interpretation of a similarity dependency

A similarity dependency $X \rightarrow Y$ holds in a table T if and only if

$$\{X\}^{\square} = \{XY\}^{\square}$$

in the pattern structure $(\mathcal{U}, (Tolerance(T), \sqcap), \theta)$

This is the same interpretation as for Functional Dependencies

Similarity Dependencies and Partition Structures: an Example

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

- With the tolerance relation $t_i \theta_m t_j \iff |t_i(m) - t_j(m)| \leq 2$,
- $ac \rightarrow d$ holds because:

$$\begin{aligned}
\{a, c\}^{\square} = \delta(a) \sqcap \delta(c) &= \{\{t_1, t_3\}, \{t_2, t_4\}\} \sqcap \{\{t_1, t_2, t_3\}, \{t_4\}\} \\
&= \{\{t_1, t_3\}, \{t_2\}, \{t_4\}\} \\
\{a, c, d\}^{\square} = \delta(a) \sqcap \delta(c) \sqcap \delta(d) &= \{a, c\}^{\square}
\end{aligned}$$

Similarity Dependencies and Partition Structures: an Example

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

- With the tolerance relation $t_i \theta_m t_j \iff |t_i(m) - t_j(m)| \leq 2$,
- $abc \rightarrow d$ holds because:

$$
\begin{aligned}
\{a, b, c\}^{\square} = \delta(a) \sqcap \delta(b) \sqcap \delta(c) &= \{\{t_1\}, \{t_3\}, \{t_2, t_4\}\} \sqcap \{\{t_1, t_2, t_3\}, \{t_4\}\} \\
&= \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}\} \\
\{a, b, c, d\}^{\square} &= \{a, b, c\}^{\square}
\end{aligned}
$$

Degenerated Multivalued Dependencies

- Jaume Baixeries, Mehdi Kaytoue and Amedeo Napoli. Characterizing Functional Dependencies in Formal Concept Analysis with Pattern Structures, Annals of Mathematics and Artificial Intelligence, 72:129–149, 2014.

Degenerated Multivalued Dependencies: Definition

Let $X \in \mathcal{U}$ and let $\overline{X} = \mathcal{U} \setminus \{X\}$.

A Degenerate Multivalued Dependency $X \to Y$ holds in a table T iif:

$$\forall t_i, t_j \in T : t_i(X) = t_j(X) \quad \Rightarrow \quad \begin{array}{l} t_i(Y) = t_j(Y) \\ \text{or} \\ t_i(\overline{XY}) = t_j(\overline{XY}) \end{array}$$

Usually, a DMVD $X \to Y$ is presented:

$$X \Rightarrow Y \mid Z$$

where $Z = \mathcal{U} \setminus XY$ and $X \cup Y \cup Z = \mathcal{U}$

Degenerated Multivalued Dependencies: Definition

Degenerate Multivalued Dependency $X \rightarrow Y$

vs

Functional Dependency $X \rightarrow Y$

$$\forall t_i, t_j \in T : t_i(X) = t_j(X) \quad \Rightarrow \quad t_i(Y) = t_j(Y)$$

or

$$t_i(\overline{XY}) = t_j(\overline{XY})$$

Degenerated Multivalued Dependencies: Definition

$a \Rightarrow b \mid cd$ holds in

id	a	b	c	d
t_1	1	3	4	1
t_2	1	3	2	3
t_3	4	6	6	2
t_4	4	5	6	2

Degenerated Multivalued Dependencies: Definition

$a \Rightarrow b \mid cd$ holds in

id	a	b	c	d
t_1	1	3	4	1
t_2	1	3	2	3
t_3	4	6	6	2
t_4	4	5	6	2

Degenerated Multivalued Dependencies: Definition

The tolerance relation $\mathcal{R}_X(T)$ in a table T induced by X is:

$$\mathcal{R}_X(T) = \{(t_i, t_j) \in T \times T \mid i < j \text{ and } t_i(X) = t_j(X) \text{ or } t_i(\overline{X}) = t_j(\overline{X})\}$$

For instance,

$$(t_1, t_2) \in \mathcal{R}_{ad}(T)$$

in:

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3

Degenerated Multivalued Dependencies: Definition

This relation is clearly reflexive and symmetric, but not necessarily transitive:

id	a	b	c	d
t_1	1	2	3	4
t_2	1	3	4	5
t_3	2	3	4	5

$$(t_1, t_2) \in \mathcal{R}_a(T) \quad \checkmark$$

$$(t_2, t_3) \in \mathcal{R}_a(T) \quad \checkmark$$

$$(t_1, t_3) \notin \mathcal{R}_a(T) \quad \text{X}$$

DMVD's and FCA: An Example

We want to compute the degenerate multivalued dependencies that hold in this table:

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

using Formal Concept Analysis

DMVD's and FCA: An Example

These are:

$a \Rightarrow b \mid cd$	$a \Rightarrow bc \mid d$	$a \Rightarrow bd \mid c$	$d \Rightarrow a \mid bc$
$d \Rightarrow ab \mid c$	$d \Rightarrow ac \mid b$	$ab \Rightarrow c \mid d$	$ac \Rightarrow b \mid d$
$ad \Rightarrow b \mid c$	$bd \Rightarrow a \mid c$	$cd \Rightarrow a \mid b$	

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

(we ommit trivial DMVD's: $X \Rightarrow Y \mid Z$, where $Y \subseteq X$ or $Z \subseteq X$)

DMVD's and FCA: An Example

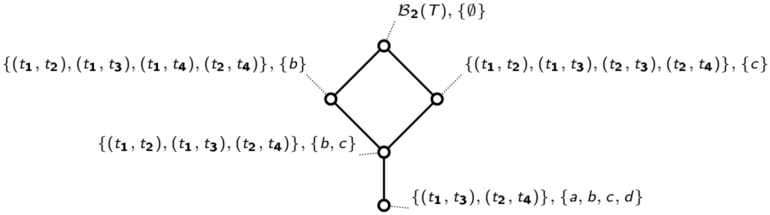
We build the Formal Context $\mathbb{K} = (\mathcal{U}, \mathcal{B}_2(T), I)$

$\mathbb{K}$	a	b	c	d
(t_1, t_2)		$\times$	$\times$	
(t_1, t_3)	$\times$	$\times$	$\times$	$\times$
(t_1, t_4)		$\times$		
(t_2, t_3)			$\times$	
(t_2, t_4)	$\times$	$\times$	$\times$	$\times$
(t_3, t_4)				

$$x \ I \ (t_i, t_j) \iff t_i(x) = t_j(x) \text{ or } t_i(\overline{x}) = t_j(\overline{x})$$

DMVD's and FCA: An Example

The concept lattice



DMVD's and FCA: An Example

We interpret a DMVD in that formal context

A DMVD $X \Rightarrow Y \mid Z$ holds in a table T if and only if

$$X' = XY' \ \cup \ XZ'$$

in the formal context $\mathbb{K} = (\mathcal{U}, \mathcal{B}_2(T), I)$

DMVD's and FCA: An Example

We interpret a DMVD in that formal context

A DMVD $X \Rightarrow Y \mid Z$ holds in a table T if and only if

$$X' = XY' \;\cup\; XZ'$$

in the formal context $\mathbb{K} = (\mathcal{U}, \mathcal{B}_2(T), I)$

REMEMBER!

A FD $X \rightarrow Y$ holds in a table T if and only if

$$X' = XY'$$

in the formal context $\mathbb{K} = (\mathcal{U}, \mathcal{B}_2(T), I)$

DMVD's and FCA: An Example

$a \Rightarrow b \mid cd$ holds because

$$a' = ab' \cup acd'$$

$$\{(t_1, t_3), (t_2, t_4)\} = \{(t_1, t_3), (t_2, t_4)\} \cup \{(t_1, t_3), (t_2, t_4)\}$$

DMVD's and FCA: An Example

$b \Rightarrow a \mid cd$ does not hold because

$$b' \neq ab' \cup bcd'$$

$$\{(t_1, t_2), (t_1, t_3), (t_1, t_4), (t_2, t_4)\} \neq \{(t_1, t_3), (t_2, t_4)\} \cup \{(t_1, t_3), (t_2, t_4)\}$$

DMVD's and Pattern Structures: An Example

We want to compute the degenerate multivalued dependencies that hold in this table:

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

using Pattern Structures

DMVD's and Pattern Structures: An Example

These are:

$$
\begin{array}{llll}
a \Rightarrow b \mid cd & a \Rightarrow bc \mid d & a \Rightarrow bd \mid c & d \Rightarrow a \mid bc \\
d \Rightarrow ab \mid c & d \Rightarrow ac \mid b & ab \Rightarrow c \mid d & ac \Rightarrow b \mid d \\
ad \Rightarrow b \mid c & bd \Rightarrow a \mid c & cd \Rightarrow a \mid b &
\end{array}
$$

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

(we ommit trivial DMVD's: $X \Rightarrow Y \mid Z$, where $Y \subseteq X$ or $Z \subseteq X$)

DMVD's and Pattern Structures: An Example

We construct the Pattern Structure:

$$(M, (D, \sqcap), \delta)$$

- M is the set of attributes $\mathcal{U}$ of the original table.
- D is the lattice of tolerance relations of the original table
- $\sqcap$ is the meet in the lattice of tolerance relations of the original table
- $\delta(X)$ is the function $\mathcal{R}_X(T)$

DMVD's and Pattern Structures: An Example

We construct the Pattern Structure $(M, (D, \sqcap), \delta)$

M is the set of attributes $\mathcal{U}$ of the original table:

$$M = \{a, b, c, d\}$$

id	a	b	c	d
t_1	1	3	4	1
t_2	4	3	4	3
t_3	1	8	4	1
t_4	4	3	7	3

DMVD's and Pattern Structures: An Example

We construct the Pattern Structure $(M, (D, \sqcap), \delta)$

The mapping of the descriptions of the attributes

$$\delta(X) : \mathcal{U} \mapsto D \text{ is } \mathcal{R}_X(T)$$

$m \in M$	$\delta(m) \in (D, \sqcap) = \mathcal{R}_m(T)$
a	$\{\{t_1, t_3\}, \{t_2, t_4\}\}$
b	$\{\{t_1, t_2, t_4\}, \{t_1, t_3\}\}$
c	$\{\{t_1, t_2, t_3\}, \{t_2, t_4\}\}$
d	$\{\{t_1, t_3\}, \{t_2, t_4\})\}$

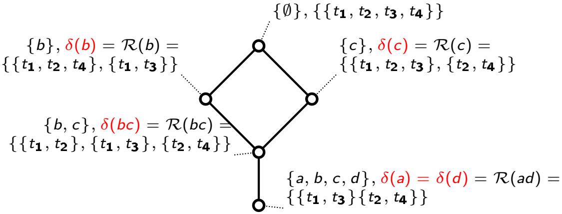
DMVD's and Pattern Structures: An Example

We can now compute the closures of sets of objects, and sets of interpretations

$$
\begin{aligned}
\{a, b\}^{\square} &= \delta(a) \sqcap \delta(b) \\
&= \{\{t_1, t_3\}, \{t_2, t_4\}\} \sqcap \{\{t_1, t_2, t_4\}, \{t_1, t_3\}\} \\
&= \{\{t_1, t_3\}, \{t_2, t_4\}\} \\
\{\{t_1, t_3\}, \{t_2, t_4\}\}^{\square} &= \{m \in M \mid \{\{t_1, t_3\}, \{t_2, t_4\}\} \sqsubseteq \delta(m)\} \\
&= \{a, b, c, d\}
\end{aligned}
$$

DMVD's and Pattern Structures: An Example

The Pattern Lattice



DMVD's and Pattern Structures: An Example

A DMVD dependency $X \Rightarrow Y \mid Z$ holds in a data table T if and only if:

$$\{X\}^{\square} = \{XY\}^{\square} \cup \{XZ\}^{\square}$$

in the partition pattern structure $(\mathcal{U}, (Tolerance(T), \sqcap), \mathcal{R}_X(T))$

DMVD's and Pattern Structures: An Example

A DMVD dependency $X \Rightarrow Y \mid Z$ holds in a data table T if and only if:

$$\{X\}^{\square} = \{XY\}^{\square} \cup \{XZ\}^{\square}$$

in the partition pattern structure $(\mathcal{U}, (Tolerance(T), \sqcap), \mathcal{R}_X(T))$

REMEMBER!

A DMVD $X \Rightarrow Y \mid Z$ holds in a table T if and only if

$$X' = XY' \cup XZ'$$

in the formal context $\mathbb{K} = (\mathcal{U}, \mathcal{B}_2(T), I)$

DMVD's and Pattern Structures: An Example

$a \Rightarrow b \mid cd$ does holds because

$$\{a\}^{\square} = \delta(a) = \{\{t_1, t_3\}\{t_2, t_4\}\}$$

$$\{ab\}^{\square} = \delta(a) \sqcap \delta(b) = \{\{t_1, t_3\}\{t_2, t_4\}\}$$

$$\{acd\}^{\square} = \delta(a) \sqcap \delta(c) \sqcap \delta(d) = \{\{t_1, t_3\}\{t_2, t_4\}\}$$

$$\{a\}^{\square} = \{ab\}^{\square} \cup \{acd\}^{\square}$$

DMVD's and Pattern Structures: An Example

$b \Rightarrow a \mid cd$ does not hold because

$$\{b\}^{\square} = \delta(b) = \{\{t_1, t_2, t_4\}\{t_1, t_3\}\}$$

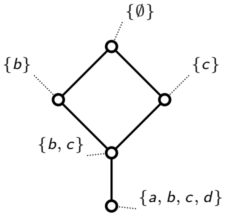
$$\{ab\}^{\square} = \delta(a) \sqcap \delta(b) = \{\{t_1, t_3\}\{t_2, t_4\}\}$$

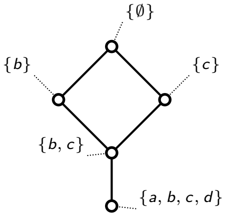
$$\{bcd\}^{\square} = \delta(b) \sqcap \delta(c) \sqcap \delta(d) = \{\{t_1, t_3\}\{t_2, t_4\}\}$$

$$\{\{t_1, t_2, t_4\}\{t_1, t_3\}\} = \{b\}^{\square} \neq \{ab\}^{\square} \cup \{acd\}^{\square} = \{\{t_1, t_3\}\{t_2, t_4\}\}$$

Relationship between Binarization and Pattern Structures

The Concept Lattice and the Pattern Lattice are isomorphic





Relationship between Binarization and Pattern Structures

What is the relationship between the formal context $\mathbb{K} = (\mathcal{U}, \mathcal{B}_2(T), I)$ and the pattern structure $(M, (D, \sqcap), \delta)$?

$$
\begin{array}{lcl}
(A, B) \text{ is a pattern concept} & & (M, (D, \sqcap), \delta) \\
 & \Leftrightarrow & \\
(B, A) \text{ is a formal concept} & & (\mathcal{B}_2(G), M, I)
\end{array}
$$

Relationship between Binarization and Pattern Structures

(A, B) is a pattern concept	$\Leftrightarrow$	$(M, (D, \sqcap), \delta)$
(B, A) is a formal concept		$(\mathcal{B}_2(G), M, I)$

	extent	intent
Formal Concept	$\{(t_1, t_2), (t_1, t_3), (t_2, t_4)\}$	$\{b, c\}$
Pattern Concept	$\{b, c\}$	$\{\{t_1, t_2\}\{t_1, t_3\}\{t_2, t_4\}\}$

Conclusion

- FCA offers a unified framework to deal with different depedencies.
- The semantics of the dependencies is embedded into the binary relation (in a Formal Context) or in the Description semi-lattice plus the δ function (in Pattern Structures).
- The interpretation and the Galois connection behind FCA handles the syntactical side.
- The results using Formal Contexts and Pattern Structures are isomorphic.

Looking Ahead

- The computation of basis for Functional Dependencies is straight forward. Can we extend it to all dependencies?
- Is FCA competitive with current algorithms for computing (minimal) basis for Functional Dependencies?
- There still more dependencies to be come: order dependencies, acyclic join dependencies.

Thanks!!

Thank you very much for your interest

Questions?