Interfaces for FCA

Bernhard Ganter

Institut für Algebra
TU Dresden
D-01062 Dresden

2008

Outline

1. Basics
   - Pairs of squares
2. Advanced
   - Dependencies
   - Rough sets
   - Background knowledge
   - Defined attributes
   - Machine learning
3. Complex
   - Implication inference as implications
   - Process exploration

Outline

1. Basics
   - Pairs of squares
2. Advanced
   - Dependencies
   - Rough sets
   - Background knowledge
   - Defined attributes
   - Machine learning
3. Complex
   - Implication inference as implications
   - Process exploration

Outline

1. Basics
   - Pairs of squares
2. Advanced
   - Dependencies
   - Rough sets
   - Background knowledge
   - Defined attributes
   - Machine learning
3. Complex
   - Implication inference as implications
   - Process exploration

Outline

A formal context . . .

A **formal context** consists of

- a set G of **objects**,
- a set M of **attributes**, and
- an **incidence relation** $I \subseteq G \times M$.

To each formal context its **concept lattice** is constructed in a natural and well known way.

. . . of pairs of squares . . .

As an example, let

- the objects be all *pairs of unit squares*,
- the attributes be
  - have a *common edge*,
  - have a *common segment*,
  - have a *common vertex*,
  - *overlap*,
  - are *disjoint*, and
  - are *parallel*,
- and the incidence defined in an obvious way.

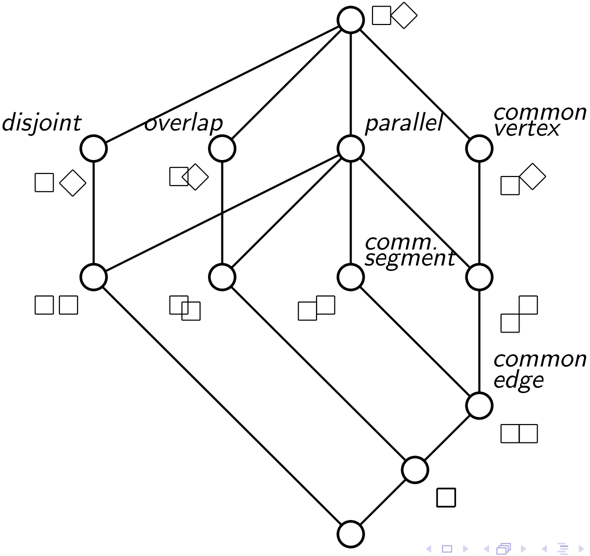
. . . with selected objects . . .

This formal context has infinitely many objects.

However, if we identify objects having the same attributes as **indiscernible**, then only finitely many classes remain. This process is called object **clarification**.

We may therefore replace the infinite object set by a finite set of representatives.

. . . and its concept lattice.



Did we select a representative object set?

Did we pick an object from each indiscernibility class?

A possible strategy for finding this out is to check whether the **logic** of the selected data set holds in general.

This gives at least a *necessary* condition for completeness:
If we find some logical expression

- that is true for all our examples,
- but is not true in general,

then our choice of examples is incomplete.

Implications

We restrict ourselves to **implications**, i.e., to expressions $A \to B$, interpreted as

if an object has all the attributes from A
then it also has all the attributes from B.

A well known theorem by Duquenne and Guigues states that each formal context has a canonical base of implications. We call this the **stem base**.

We may use the stem base for checking completeness.

Stem base of the example set

- common edge $\longrightarrow$ parallel, common vertex, common segment
- common segment $\longrightarrow$ parallel
- parallel, common vertex, common segment $\longrightarrow$ common edge
- overlap, common vertex $\longrightarrow$ parallel, common segment, common edge
- overlap, parallel, common segment $\longrightarrow$ common edge, common vertex
- overlap, parallel, common vertex $\longrightarrow$ common segment, common edge
- disjoint, common vertex $\longrightarrow \bot$
- disjoint, parallel, common segment $\longrightarrow \bot$
- disjoint, overlap $\longrightarrow \bot$

Two of the implications do not hold in general

- common edge $\longrightarrow$ parallel, common vertex, common segment
- common segment $\longrightarrow$ parallel
- parallel, common vertex, common segment $\longrightarrow$ common edge
- overlap, common vertex $\longrightarrow$ parallel, common segment, common edge
- overlap, parallel, common segment $\longrightarrow$ common edge, common vertex
- overlap, parallel, common vertex $\longrightarrow$ common segment, common edge
- disjoint, common vertex $\longrightarrow \perp$
- disjoint, parallel, common segment $\longrightarrow \perp$
- disjoint, overlap $\longrightarrow \perp$

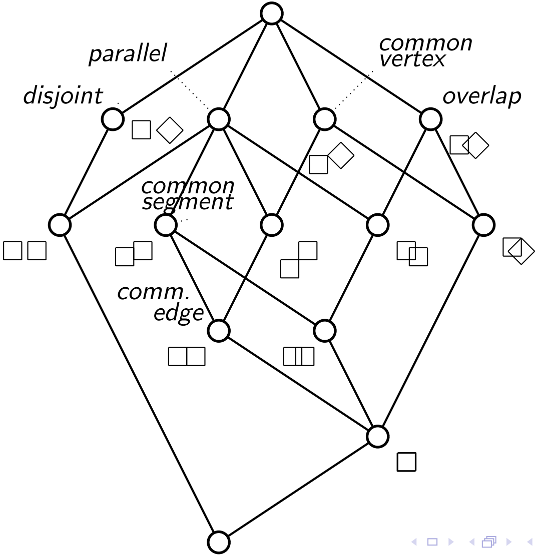
Conterexamples for the two implications

- overlap, common vertex $\rightarrow$ parallel, common segment, common edge
- Counterexample: 
- overlap, parallel, common segment $\rightarrow$ common edge, common vertex
- Counterexample:

Conterexamples for the two implications

- overlap, common vertex → parallel, common segment, common edge
- Counterexample: 
- overlap, parallel, common segment → common edge, common vertex
- Counterexample: 

A better choice of examples



The lesson

We have learnt that

> *computing the stem base is a nontrivial and useful tool for checking completeness of data sets.*

Is there a business idea waiting for exploitation?
Can the tool be applied to other kinds of situations, easily?

Outline

1. Basics
   - Pairs of squares
2. Advanced
   - Dependencies
   - Rough sets
   - Background knowledge
   - Defined attributes
   - Machine learning
3. Complex
   - Implication inference as implications
   - Process exploration

Relational data

Often data comes as an "EXCEL"-style table, which may be understood as a relation or, more conceptually formulated, as a many-valued context

$$(G, M, W, I).$$

The rows correspond to objects, the columns to (many-valued) attributes, and the entries to attribute values.

Incidence $(g, m, w) \in I$ is read as "the value for object g of attribute m is w", and is abbreviated as $m(g) = w$.

Indiscernibility

Two objects g, h may be considered to be **insdiscernible** in a given many-valued context (G, M, W, I) if $m(g) = m(h)$ holds for all attributes $m \in M$.

Indiscernibility grows when the attribute set is shortened, i.e., when we select some subset $M_0 \subseteq M$ and work with the shortened many-valued context

$$(G, M_0, W, I \cap G \times M_0 \times W).$$

Reducing the attribute set without increasing indiscernibility is of interest in **rough set analysis**.

Functional dependencies

Instead of implications, here **functional dependencies** often are studied. There is, however, a close connection.

Let (G, M, W, I) be a many-valued context. Consider to formal context

$$(G \times G, M, J),$$

where

$$(g, h)\ J\ m :\Longleftrightarrow\ m(g) = m(h).$$

The implications of the latter are precisely the functional dependencies of the first.

A lattice of dependencies

The formal context

$$(G \times G, M, J), \quad (g, h)\, J\, m :\Longleftrightarrow\ m(g) = m(h)$$

has, of course, a concept lattice. What do the formal concepts stand for?

Concept extents are subsets of $G \times G$ and therefore relations on G. In fact, they are equivalence relations. More precisely, the extents are precisely the indiscernibility relations that can be obtained by shortening the attribute set.

Concept intents are the maximal attribute sets corresponding to a given indiscernibility relation ("Coreducts").

Data base dependencies

The theory of relational data bases also studies oher kinds of dependencies, such as “tuple generating dependencies”.

In his Ph.D. thesis, Jaume Baixeries has investigated how such dependencies may be described with methods of FCA, in particular, how a formal context can be associated to a data base table for each dependency type.

This offers a possible generalization of the “rough set” approach.

Scaling

With FCA's **conceptual scaling** in mind, another general notion of dependencies suggests itself.

Functional dependencies are closely related to a **nominal** interpretation of the many-valued attributes.

Other scalings may lead to other dependencies. Such have been investigated by R. Wille, myself, and others.

Linguistic variables

Lotfi Zadeh has suggested **linguistic variables** to encode non-numerical ("unprecise") data.

In terms of FCA these may be understood as ordinally scaled many-valued attributes.

Ordinal dependencies can be introduced in the obvious manner. They are in 1-1-correspondene with the implications of the formal context

$$(G \times G, M, K), \quad (g, h)\ K\ m :\iff m(g) \leq m(h).$$

Non-symmetric indiscernibility

The formal concepts of

$$(G \times G, M, K), \quad (g, h)\ K\ m :\iff m(g) \leq m(h)$$

can naturally be interpreted:

Concept extents are the ordinal indiscernibility relations that can be obtained by shortening the attribute set. These relations are quasi-orders.

Concept intents are the maximal attribute sets corresponding to a given indiscernibility relation.

Rough set approximations

Zdzisław Pawlak's original definition of a rough set is based on an indiscernibility equivalence relation on an "universe" U.

The **definable sets** are defined as those that can be written as a union of indiscernibility equivalence classes.

The **rough set** approximating any given subset $A \subseteq U$ is a pair of definable sets

$$(\underline{R}(A), \overline{R}(A)),$$

called **lower** and **upper approximation** of A.

Rough sets as intervals

The approximating pairs $(\underline{R}(A), \overline{R}(A))$ may be understood as intervals in the boolean lattice of definable sets.

Thereby, rough set analysis offers an *interval arithmetic for sets*, based on the indiscernibility equivalence relation.

With some simplifications the algebra of rough sets is the (Stone) lattice of intervals of a boolean algebra.

Generalized rough set approximations

In a recent paper we have systematically studied possible generalizations of the rough set approach, replacing $\underline{R}(A)$ and $\overline{R}(A)$ by an arbitrary kernel-closure operator pair.

This can nicely be treated using the FCA theory of subdirect product constructions, taking advantage of the P-**fusion** construction for formal contexts.

A full characterizaion could be achieved. It was also clarified under which conditions the algebra of generalized rough sets is an algebra of lattice intervals.

Quasi order ideals

It turns out that the nice case is exactly that of a not necessarily symmetric indiscernibility quasi order, as it is natural in the presence of linguistic variables.

The definable sets are precisely the **quasi order ideals**. The lattice of definable sets is distributive, but not necessarily boolean.

The rough set approximations then again are just the intervals of the lattice of definable sets. We thus obtain an interval arithmetic for linguistic variables.

Implications and scaling

There is an obvious answer to the question what the implications of a many-valued context are:

The implications depend on the scaling, and are automatically given once a scaling is choosen. They are simply the implications of the derived one-valued context.

It is indeed that simple. Nevertheless some extra considerations are required, as a simple example shows.

An unpleasant example: Driving test

	theory		driving		license	
	pass	fail	pass	fail	pass	fail
1	×		×		×	
2	×			×		×
3		×	×			×
4		×		×		×

We expect that the stem base, which is irredundant, sound and complete, expresses that

$$\text{license} = \text{pass} \quad \leftrightarrow \quad \text{theory} = \text{pass } \textit{and} \text{ practice} = \text{pass}.$$

. . . and its stem base

driving = fail	$\rightarrow$	license = fail
theory = fail	$\rightarrow$	license = fail
license = fail, driving = pass	$\rightarrow$	theory = fail
license = fail, theory = pass	$\rightarrow$	driving = fail
driving = pass, theory = pass	$\rightarrow$	license = pass
license = pass	$\rightarrow$	driving = pass, theory = pass
license = fail, theory = fail, driving = pass, driving = fail	$\rightarrow$	$\perp$
license = fail, theory = fail, theory = pass, driving = fail	$\rightarrow$	$\perp$

Something must be wrong?

Background knowledge

The reason why the driving test stem base is so complicated is that the stem base algorithm does not "know" that PASS and FAIL are negations of each other.

This shows the necessity to include **background knowledge**, which may be non-implicational.

One frequent instance is that the background knowledge encodes the scaling information of a many-valued context.

Modified inference

Including background knowledge requires a modified inference mechanism for implications.

In general, this would make the inference problem intractable.

But it can be shown that for a *fixed amount* of non-implicational background knowledge, implication inference remains tractable.

This typically applies to scaling-induced background knowledge.

Relational contexts

In many situations it is natural to consider formal contexts with an additional relational structure on the object set.

This can be modelled using **relational contexts** or **power context families.**

Prominent such cases are **Description Logics**, where the relations are binary and are called **roles**.

Contributions were made by Baader, Rudolph, Sattler, Sertkaya and others.

DL attribute logic

Special aspects of description logic contexts and their implications are:

- Background knowledge must be taken into account. Here the background knowlede comes from the attribute definitions.
- DL contexts typically are *partial*, i.e., incidence is usually unknown for many object–attribute pairs.
- The number of potentially definable attributes usually is infinite, even for finite relational contexts.
- Implication inference should be left to the powerful DL reasoning algorithms, by default.

Learning data

Suppose that your formal context (G, M, I) is divided

$$G = G_+ \mathbin{\dot{\cup}} G_-$$

into a set G_+ of **positive examples** and a set G_- of **negative examples**.

It is desired to find a characterization of the positive examples in terms of the formal context.

This is easy when G_+ is a concept extent, but that is an exceptionally easy case.

Hypotheses

Sergei Kuznetsov, using ideas from Machine Learning, has introduced (positive) **hypotheses** as concept intents, the extents of which consist entirely of positive examples.

One may restrict to minimal such hypotheses.

The goal then is to cover as many as possible of the positive examples by positive hypotheses.

Distributivity

Explaining a set of positive examples by hypotheses uses and "**or**" construction:

> ... object g is a positive example because it satisfies hypothesis H_1 **or** hypothesis H_2 **or** ....

A more careful analysis yields the following chain:

or → distributive → order ideals → linguistic variables → non-symmetric indiscernibility → ...

Outline

The Armstrong rules

The stem base is complete in the following sense: Each implication that holds in a given formal context can be inferred from the stem base.

The inference rules are the **Armstrong rules**, well known from the theory of functional dependencies.

This is easy and straightforward. Nevertheless, we may derive these rules in a different way, using a formal context, as was recently shown by Johannes Wollbold.

A formal context for implication inference

Fix an attribute set M, and construct a formal context as follows:

- The **objects** are all formal contexts with attribute set M, up to object clarification and isomorphism.
- The **attributes** are all implications $A \to B$, $A, B \subseteq M$.
- A formal context is incident with an implication iff the implication holds in that context.

Formally

For short, we get

$$(\mathfrak{P}(\mathfrak{P}(M)), \mathfrak{P}(M) \times \mathfrak{P}(M), \models).$$

The implications of this context correspond to implication inference on M.

This context has a horrilbly large stem base, even for small sets M.

It needs to be **folded**.

Process Howtos

Several authors (e.g., Rudolph, Wolff, Wollbold) made attempts to model processes conceptually. Wollbold works with biological data (gene expressions).

Is it possible to apply the stem base technique to such data? What are objects, what are attributes?

There are some formal models available (labelled transition systems, boolean networks, fluent calculus), but it is not obvious how to interprete these in contextual attribute logic.

Change as object

One possible approach is to introduce **states** having attributes, modelling the static perspective.

In addition, **transitions** are considered that change from one state to another, requiring **preconditions** and resulting in **postconditions**.

Using these transitions as objects leads to a formal context with doubled attribute set (pre/post), the implications of which can be studied.

Finding the inference rules for these implications is nontrivial.