Adding Background Knowledge to Formal Concept Analysis via Attribute Dependency Formulas

Radim Belohlavek
Binghamton University — SUNY
Binghamton, NY 13902, U. S. A.
rbelohla@binghamton.edu

Vilem Vychodil
Binghamton University — SUNY
Binghamton, NY 13902, U. S. A.
vychodil@binghamton.edu

ABSTRACT

We present a way to add user's background knowledge to formal concept analysis. The type of background knowledge we deal with relates to relative importance of attributes in the input data. We introduce AD-formulas which represent this type of background knowledge. The background knowledge serves as a constraint. The main aim is to make extraction of clusters from the input data more focused by taking into account the background knowledge. Particularly, only clusters which are compatible with the background knowledge are extracted from data. As a result, the number of extracted clusters becomes smaller, leaving out non-interesting clusters. We present illustrative examples and results on entailment of background knowledge such as efficient testing of entailment and a complete systems of deduction rules.

Categories and Subject Descriptors

I.2.3 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*relation systems*; H.2.8 [**Database Management**]: Database Applications—*data mining*; I.5.3 [**Artificial Intelligence**]: Clustering; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic

General Terms

Algorithms, Theory, Human Factors

Keywords

formal concept analysis, background knowledge, attribute dependencies, entailment, completeness

1. INTRODUCTION AND PRELIMINARIES

1.1 Problem Description and Paper Content

The paper presents a contribution to formal concept analysis (FCA). We investigate a way to extend the basic setting of FCA by taking into account a particular type of user's background knowledge regarding the input data. The main benefit of adding the background knowledge is that instead of extracting all clusters, the number of which can be quite large, our approach allows to extract only those clusters which are compatible with the background knowledge. As a result, the user gets only "interesting" clusters (those compatible with background knowledge) instead of being overwhelmed by a large number of both "interesting" and "non-interesting" clusters. In addition to that, our approach lends itself to theoretical analysis. As an example, we present results related to reasoning with background knowledge. We show that entailment can be efficiently tested and that there is a complete set of Armstrong-like inference rules for our type of background knowledge. As a result, for instance, redundancy can be removed from the background knowledge specified by a user.

The paper is organized as follows. Sections 1.2 and 1.3 present preliminaries on formal concept analysis and related approaches, respectively. Section 2 describes our approach. First, we present the rationale and informal description. Second, we present technical details and results regarding entailment, its testing, and Armstrong-like rules. Section 3 provides examples. Section 4 presents conclusions and outlines some directions of future research.

1.2 Preliminaries from FCA

In this section, we survey basic notions from formal concept analysis (FCA). FCA is a method of knowledge extraction from data tables describing relationships between objects and attributes [3, 7]. The input data table is represented by a triplet $\langle X, Y, I\rangle$, where X and Y are non-empty sets of objects (table rows) and attributes (table columns), and I is a binary relation between X and Y indicating whether object $x \in X$ has attribute $y \in Y$ or not. In the former case, $\langle x, y\rangle \in I$ and the corresponding table entry contains 1, in the latter case, $\langle x, y\rangle \notin I$ and the entry contains 0. An example of such table is in Table 1 which we use for illustration in our paper. A (*formal*) *concept* in $\langle X, Y, I\rangle$ is a pair $\langle A, B\rangle$ of a set $A \subseteq X$ of objects (so-called *extent*) and a set $B \subseteq Y$ of attributes (so-called *intent*) such that A is the set of all objects which have all attributes from B, and B is the set of all attributes shared by all objects from A. This can be expressed using arrow operators $^{\uparrow}$ and $^{\downarrow}$ by $A^{\uparrow} = B$ and $B = A^{\downarrow}$ where

$$A^{\uparrow} = \{y \in Y \mid \text{for each } x \in A\colon \langle x, y\rangle \in I\},$$
$$B^{\downarrow} = \{x \in X \mid \text{for each } y \in B\colon \langle x, y\rangle \in I\}.$$

Alternatively, formal concepts can be described as maximal rectangles in the table which contain 1s. Formal concepts

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil

Table 1: Input data table

	genus				habitat				size			fur		color			
	Acinonyx	Felis	Leptailurus	Panthera	Africa	America	Asia	Europe	small	medium	large	stripes	spots	black	sandy	white	yellow
Cheetah	1	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	1
Cougar	0	0	0	1	0	1	0	0	0	0	1	0	0	0	1	0	0
Jaguar	0	0	0	1	0	1	0	0	0	0	1	0	1	1	0	0	1
Lion	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	0	0
Panther	0	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	1
Serval	0	0	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1
Tiger	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	1	1
Wildcat	0	1	0	0	1	1	1	1	1	0	0	1	1	1	1	0	1

can be partially ordered by a subconcept-superconcept hierarchy defined by $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ iff $A_1 \subseteq A_2$ (or, equivalently, $B_2 \subseteq B_1$). The set $\mathcal{B}(X, Y, I)$ of all formal concepts in $\langle X, Y, I\rangle$, i.e.

$$\mathcal{B}(X, Y, I) = \{\langle A, B\rangle \mid A \subseteq X,\ B \subseteq Y,\ A^{\uparrow} = B,\ B = A^{\downarrow}\}$$

equipped with $\leq$ is called a *concept lattice* of $\langle X, Y, I\rangle$. $\mathcal{B}(X, Y, I)$ represents a collection of hierarchically ordered clusters extracted from the input data. Various applications of FCA can be found in [3] and in the references therein.

1.3 Related Approaches

The idea of using background knowledge appears in various forms in artificial intelligence. Using background knowledge for constraining purposes has recently been studied in several papers, see e.g. [4, 12]. In formal concept analysis, particularly, a type of background knowledge has been studied in [6, 11]. There, the authors use attribute implications (see later) for the purpose of attribute exploration and both the type of the background knowledge and the aims are different from ours. The present paper is a continuation of [1] where the idea of attribute dependency formulas and their role in serving as a constraint was introduced.

2. AD-FORMULAS, ENTAILMENT, AND ARMSTRONG-LIKE RULES

2.1 AD-formulas

Motivation In the basic setting of FCA, the input data consists of a table $\langle X, Y, I\rangle$ describing the objects, attributes, and their relationship. The concept lattice is then computed from this data. Such data does not capture background knowledge which a user may have regarding the data. Extracting formal concepts and the concept lattice associated to $\langle X, Y, I\rangle$ without taking the background knowledge into account means, in fact, ignoring the background knowledge. This can result in extraction of a large number of formal concepts including those which seem artificial to the user because they are not congruent with his background knowledge.

A particular type of background knowledge which we deal with in this paper is relative importance of attributes. Such type of background knowledge is commonly used in human categorization/clustering. For instance, when categorizing books for the purpose of inclusion in a sales catalogue, one might consider the field subject of a book more important than the type of book. Accordingly, we expect to form categories of books based on the subject, such as "Engineering", "Computer Science", "Mathematics", "Biology", etc., and only after that, within these categories, we might want to form smaller categories based on the type of reader such as "Engineering/textbook", "Engineering/research monograph", etc. In such a case, our background knowledge tells that attributes describing the subject ("Engineering", "Computer Science", ...) are more important than attributes describing the type of book ("textbook", "research monograph"). The background knowledge depends on the purpose of categorization. For a different purpose, it can be appropriate to use different type of background knowledge. For instance, one could consider the type of book more important than the subject. Correspondingly, we would get categories "textbook", "research monograph" and their subcategories "textbook/Engineering", "textbook/Computer Science", etc. Therefore, while the input data is given (books and their attributes), the background knowledge which guides the categorization is purpose dependent.

The relative importance of attributes serves as a constraint in categorization/clustering. Namely, it excludes potential clusters which do not respect the background knowledge. For instance, with subject more important than type, category "textbook" consisting of all textbooks (irrelevant of the subject) does not exist (is not formed in the process of categorization), because it is not congruent with background knowledge (does not satisfy the corresponding constraint saying that subject is more important than type). Contrary to that, categories "Engineering", "Engineering/textbook", "Computer Science", "Computer Science/textbook" are congruent with the background knowledge.

Background knowledge can not only eliminate categories which are not suitable for a given purpose, it also can eliminate unnatural categories. As an example, not taking into account that book binding is less important than book subject and book type, one would end up with categories "paperback" and "hardbound" which, however logically correct, do not make much sense in a useful categorization of books.

The concept of AD-formula The informal ideas described above can be approached in the framework of FCA as follows.

Definition 1. An *attribute-dependency formula* (shortly, an AD-formula) over Y is an expression

$$A \sqsubseteq B,$$

where $A, B \subseteq Y$. $A \sqsubseteq B$ *is true in* $M \subseteq Y$, written $M \models A \sqsubseteq B$, if we have:

$$\text{if } A \cap M \neq \emptyset \text{ then } B \cap M \neq \emptyset. \tag{1}$$

A formal concept $\langle C, D\rangle \in \mathcal{B}(X, Y, I)$ satisfies $A \sqsubseteq B$ iff $D \models A \sqsubseteq B$. □

Example 1. Let us take a closer look at Table 1 which we use as our working example and to which we return in detail in Section 3. Table 1 describes selected felines (objects) and their characteristics (attributes). Each row of the table is a record of characteristics of a species of felines. The attributes can be divided into four groups: a biological genus (Acinonyx, Felis, Leptailurus, and Panthera), natural

habitat of the species (attributes representing continents), size of its body (small/medium/large), fur pattern (fur with stripes and/or spots), and color in which the species may appear. Thus, the input table consists of 8 objects and 17 attributes. A table like this may be constructed by a biologist who wishes to use the information contained in the table to form clusters of species with common properties. This task is, in fact, an application of formal concept analysis.

The corresponding concept lattice $\mathcal{B}(X, Y, I)$ contains, among others, the formal concept $\langle C, D\rangle$ with

$$C = \{\text{Cheetah, Jaguar, Panther, Serval, Tiger, Wildcat}\},$$
$$D = \{\text{yellow}\}.$$

Note that this formal concept represents a category/cluster of all felines with yellow fur. However, for a biologist, such a category may seem unnatural (there is no such a concept as "felines with yellow fur" for a biologist). This is because whenever the biologist considers the color of fur for the purpose of categorization, he always considers other attributes which are more important, such as "being a species which belongs to the *Panthera* genus". That is, the biologist considers genus more important than color. This background knowledge can be expressed by means of the AD-formula

$$\{\text{black, sandy, white, yellow}\} \sqsubseteq \{\text{Acinonyx, Felis, Leptailurus, Panthera}\}. \quad (2)$$

The above formal concept $\langle C, D\rangle$ does not satisfy this AD-formula because

$$\{\text{black, sandy, white, yellow}\} \cap \{\text{yellow}\} \neq \emptyset$$

but

$$\{\text{Acinonyx, Felis, Leptailurus, Panthera}\} \cap \{\text{yellow}\} = \emptyset.$$

That is, while attribute "yellow" is used in the description (intent) D, of the formal concept $\langle C, D\rangle$, none of the more important attributes specified by AD-formula (2) is. On the other hand, formal concept $\langle C, D\rangle$ with

$$C = \{\text{Jaguar, Panther, Tiger}\},\ D = \{\text{Panthera, yellow}\}.$$

satisfies (2). This formal concept corresponds to the category of species within genus *Panthera* which can appear yellow. □

The set of all formal concepts from $\mathcal{B}(X, Y, I)$ which satisfy a given set T of AD-formulas is denoted by $\mathcal{B}_T(X, Y, I)$, i.e.

$$\mathcal{B}_T(X, Y, I) = \{\langle C, D\rangle \in \mathcal{B}(X, Y, I) \mid \text{for every } A \sqsubseteq B \in T : D \models A \sqsubseteq B\}.$$

Remark 1. (i) In [1], AD-formulas were introduced as expressions of the form

$$y \sqsubseteq y_1 \sqcup \cdots \sqcup y_n,$$

i.e., $\{y\} \sqsubseteq \{y_1, \ldots, y_n\}$ in the present notation. The present extension to formulas of the $\{z_1, \ldots, z_m\} \sqsubseteq \{y_1, \ldots, y_n\}$ is not essential. Namely, as can be easily shown, a formal concept $\langle C, D\rangle$ satisfies $\{z_1, \ldots, z_m\} \sqsubseteq \{y_1, \ldots, y_n\}$ if and only if $\langle C, D\rangle$ satisfies every $\{z_i\} \sqsubseteq \{y_1, \ldots, y_n\}$.

(ii) In an AD-formula $A \sqsubseteq B$, A and B are usually collections of attributes of the same kind such as in (2). This makes is possible to attach an apt meaning to $A \sqsubseteq B$ such as "color is less important than genus".

(iii) (1) is just the condition of validity of dependencies considered in Knowledge Spaces [5] where M is interpreted as a set of questions an individual can answer and $A \sqsubseteq B$ being true in M means that if that individual fails in answering all questions from A then he fails in answering all questions from B. Our aims and the subsequent development of AD-formulas are different from those in [5]. □

In [1], we presented results and examples related to expressive capability of AD-formulas. In the subsequent sections, we pay attention to entailment and inference over AD-formulas.

2.2 Entailment and its efficient checking

Background knowledge specified by a user by means of AD-formulas may be redundant. For instance, suppose the background knowledge consists of (2) and AD-formulas

$$\{\text{black, sandy, white, yellow}\} \sqsubseteq \{\text{Africa, America, Asia, Europe}\} \quad (3)$$

$$\{\text{Africa, America, Asia, Europe}\} \sqsubseteq \{\text{Acinonyx, Felis, Leptailurus, Panthera}\}. \quad (4)$$

That is, (2), (3), and (4) say "color is less important than genus", "color is less important than habitat", and "habitat is less important than genus". Intuitively, (2) is redundant because it is entailed by (3) and (4). One may wish to remove such redundancy because it makes the description of background knowledge less comprehensible. Below, in addition to making the notion of entailment precise, we present results which lead to an efficient algorithm for testing entailment and removal of redundancy.

Definition 2. A set $M \subseteq Y$ is called a *model* of a set T of AD-formulas if, for each $A \sqsubseteq B \in T$, $M \models A \sqsubseteq B$. Let $\mathrm{Mod}(T)$ denote the set of all models of T. $A \sqsubseteq B$ *semantically follows from* T (T *semantically entails* $A \sqsubseteq B$) if, for each $M \in \mathrm{Mod}(T)$, $M \models A \sqsubseteq B$. □

The following theorem shows a first technical insight.

THEOREM 1. *Let T be a set of AD-formulas. Then* $\mathrm{Mod}(T)$ *is an interior system, i.e.* $\mathrm{Mod}(T)$ *is closed under arbitrary unions.*

By virtue of Theorem 1, for each set T of AD-formulas, we can consider the associated interior operator $I_T : 2^Y \to 2^Y$ defined by $I_T(M) = \bigcup\{N \in \mathrm{Mod}(T) \mid N \subseteq M\}$. Clearly, we have $\mathrm{Mod}(T) = \{M \subseteq Y \mid M = I_T(M)\}$. Furthermore, for every $M \subseteq Y$, $I_T(M)$ is the greatest model of T which is included in M. The following algorithm shows a way to compute $I_T(M)$ given T and M.

Algorithm 1.

Input: set T of AD-formulas, $M \subseteq Y$
Output: $I_T(M)$

set N to M
while there is $A \sqsubseteq B \in T$ such that $N \not\models A \sqsubseteq B$:
  choose $A \sqsubseteq B \in T$ and $y \in A$ such that $y \in N$ and $B \cap N = \emptyset$
  remove y from N
return N

The next theorem shows a crucial property. Namely, testing entailment can be performed by checking validity in a single model: T entails $A \sqsubseteq B$ iff $A \sqsubseteq B$ is true in the greatest model of T which is contained in the complement $\overline{B}$ of B.

Note that testing entailment using a single model is known in other areas too, see e.g. [7] for attribute implications or [9] for logic programming.

THEOREM 2. *Let T be a set of AD-formulas, $A \sqsubseteq B$ be an AD-formula. Then the following are equivalent.*

(i) $T \models A \sqsubseteq B$,
(ii) $I_T(\overline{B}) \models A \sqsubseteq B$,
(iii) $A \cap I_T(\overline{B}) = \emptyset$.

Thus, Algorithm 1 and Theorem 2 give us the following algorithm:

Algorithm 2.

Input: set T of AD-formulas and AD-formula $A \sqsubseteq B$
Output: *true* if $T \models A \sqsubseteq B$, *false* otherwise.

compute $I_T(\overline{B})$ *using Algorithm 1*
if $A \cap I_T(\overline{B}) = \emptyset$:
  return true
else:
  return false

2.3 Complete systems of deduction rules

The next issue related to entailment and reasoning with AD-formulas is the question of whether there is a complete system of deduction rules. We present a positive answer by showing a system of Armstrong-like rules. We use deduction rules of the form

$$\frac{A_1 \sqsubseteq B_1, \ldots, A_n \sqsubseteq B_n}{A \sqsubseteq B}. \tag{5}$$

These rules will be used in proofs in the usual way. That is, having AD-formulas which match the "input part" of the rule, i.e. $A_1 \sqsubseteq B_1, \ldots, A_n \sqsubseteq B_n$, we can infer, in a single step, the AD-formula corresponding to the "output part", i.e. $A \sqsubseteq B$. In particular, we will use the following system of deduction rules:

$(\overline{\text{Ref}})$ $\overline{A \sqsubseteq A}$,

$(\overline{\text{Wea}})$ $\dfrac{A \sqsubseteq B}{A \sqsubseteq B \cup C}$,

$(\overline{\text{Cut}})$ $\dfrac{A \sqsubseteq B,\ C \sqsubseteq A \cup D}{C \sqsubseteq B \cup D}$,

for each $A, B, C, D \subseteq Y$. The notions of a proof and provability are defined the usual way. That is, a *proof* of an AD-formula $A \sqsubseteq B$ from a set T of AD-formulas is a sequence $\varphi_1, \ldots, \varphi_n$ of AD-formulas such that $\varphi_n = A \sqsubseteq B$ and each φ_i either is from T or can be inferred from some preceding formulas φ_j, $j < i$, using some of the deduction rules $(\overline{\text{Ref}})$–$(\overline{\text{Cut}})$. An AD-formula $A \sqsubseteq B$ is *provable* from a set T of AD-formulas if there is a proof of $A \sqsubseteq B$ from T; in this case, we write $T \vdash A \sqsubseteq B$. The following assertion is a completeness theorem for reasoning with AD-formulas.

THEOREM 3 (COMPLETENESS). $(\overline{\text{Ref}})$–$(\overline{\text{Cut}})$ *is a sound and complete system of deduction rules. That is, for any set T of AD-formulas and an AD-formula $A \sqsubseteq B$, we have*

$$T \vdash A \sqsubseteq B \quad \textit{iff} \quad T \models A \sqsubseteq B.$$

In words, $A \sqsubset B$ is entailed by T iff $A \sqsubset B$ can be obtained from T by rules $(\overline{\text{Ref}})$–$(\overline{\text{Cut}})$.

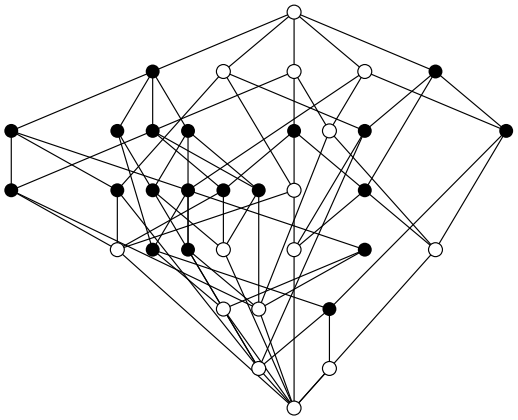

Figure 1: Hierarchy of All Conceptual Clusters

PROOF. Sketch (due to lack of space): Let $T \vdash A \sqsubseteq B$ and let $\varphi_1, \ldots, \varphi_n$ be the proof of $A \sqsubseteq B$. It can be shown that a sequence $\varphi_1, \ldots, \varphi_n$ of AD-formulas is a proof from T using rules (R), ..., iff the corresponding sequence $(\varphi_1)^{\text{AI}}, \ldots, (\varphi_n)^{\text{AI}}$ of attribute implications is a proof from T^{AI} using rules $(\text{R})^{\text{AI}}$, ... Here, $(\cdots)^{\text{AI}}$ denotes replacing all AD-formulas in $(\cdots)$ by the corresponding attribute implications [7]. For example, $(A \sqsubseteq B)^{\text{AI}} = B \Rightarrow A$, $(\overline{\text{Ref}})^{\text{AI}}$, $(\overline{\text{Wea}})^{\text{AI}}$, $(\overline{\text{Cut}})^{\text{AI}}$ become

$$\frac{}{A \Rightarrow A}, \qquad \frac{B \Rightarrow A}{B \cup C \Rightarrow A}, \qquad \frac{B \Rightarrow A,\ A \cup D \Rightarrow C}{B \cup D \Rightarrow C},$$

etc. Now, $(\overline{\text{Ref}})^{\text{AI}}$, $(\overline{\text{Wea}})^{\text{AI}}$, $(\overline{\text{Cut}})^{\text{AI}}$ are the well-known Armstrong rules of reflexivity, weakening, and cut, respectively, see [10], which are known to be complete w.r.t. semantics of attribute implications (alternatively, one can use the semantics of functional dependencies), see [10] and [7]. Therefore, $(\varphi_1)^{\text{AI}}, \ldots, (\varphi_n)^{\text{AI}}$ being a proof of $(A \sqsubseteq B)^{\text{AI}}$ from T^{AI} using $(\overline{\text{Ref}})^{\text{AI}}$, $(\overline{\text{Wea}})^{\text{AI}}$, $(\overline{\text{Cut}})^{\text{AI}}$ is equivalent to $T^{\text{AI}} \models (A \sqsubseteq B)^{\text{AI}}$. The latter can be shown to be equivalent to $T \models A \sqsubseteq B$. □

Note that due to the relationships to attribute implications, we can automatically retrieve further sound deduction rules over AD-formulas from the well-known rules for attribute implications (or, equivalently, functional dependencies) such as

$(\overline{\text{Add}})$ $\dfrac{B \sqsubseteq A,\ C \sqsubseteq A}{B \cup C \sqsubseteq A}$,

$(\overline{\text{Pro}})$ $\dfrac{A \cup B \sqsubseteq C}{A \sqsubseteq C}$,

$(\overline{\text{Tra}})$ $\dfrac{A \sqsubseteq B,\ B \sqsubseteq C}{A \sqsubseteq C}$.

3. EXAMPLES

Consider again the data from Table 1. The corresponding concept lattice $\mathcal{B}(X, Y, I)$ contains 35 conceptual clusters (formal concepts) and is depicted in Figure 1. As shown in Example 1, not including any background knowledge results in having formal concepts such as the one corresponding to category of felines with yellow fur. Suppose we impose constraints by adding background knowledge using the following

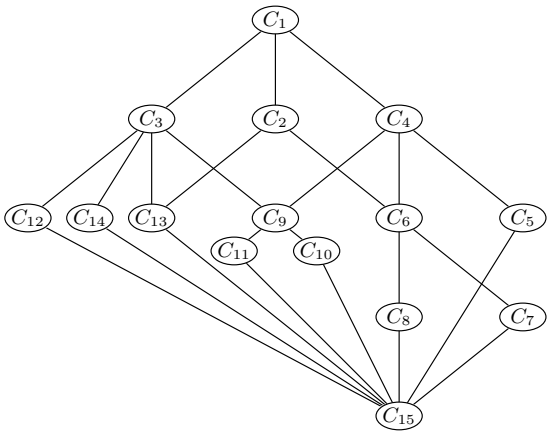

Figure 2: Constrained Hierarchy of Clusters I

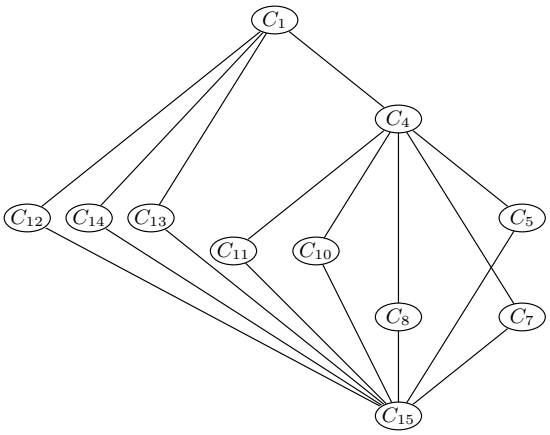

Figure 3: Constrained Hierarchy of Clusters II

AD-formulas:

$$\{\text{stripes, spots, black, sandy, white, yellow}\} \sqsubseteq \{\text{small, medium, large}\},$$
$$\{\text{small, medium, large}\} \sqsubseteq \{\text{Acinonyx, Felis, Leptailurus, Panthera}\},$$
$$\{\text{small, medium, large}\} \sqsubseteq \{\text{Africa, America, Asia, Europe}\}.$$

They say that fur color and pattern are less important than size, size is less important than genus, and size is less important than habitat. No preference is asserted between habitat and genus, nor between fur color and pattern. The corresponding constrained concept lattice $\mathcal{B}_T(X, Y, I)$ consists of 15 conceptual clusters (the names of attributes are abbreviated):

$C_1 = \langle X, \emptyset \rangle$,
$C_2 = \langle \{\text{Cougar, Jaguar, Wildcat}\}, \{\text{Am}\} \rangle$,
$C_3 = \langle \{\text{Cheetah, Lion, Panther, Serval, Wildcat}\}, \{\text{Af}\} \rangle$,
$C_4 = \langle \{\text{Cougar, Jaguar, Lion, Panther, Tiger}\}, \{\text{Pa}\} \rangle$,
$C_5 = \langle \{\text{Tiger}\}, \{\text{Pa, As, la, st, wh, ye}\} \rangle$,
$C_6 = \langle \{\text{Cougar, Jaguar}\}, \{\text{Pa, Am, la}\} \rangle$,
$C_7 = \langle \{\text{Cougar}\}, \{\text{Pa, Am, la, sa}\} \rangle$,
$C_8 = \langle \{\text{Jaguar}\}, \{\text{Pa, Am, la, sp, bl, ye}\} \rangle$,
$C_9 = \langle \{\text{Lion, Panther}\}, \{\text{Pa, Af}\} \rangle$,
$C_{10} = \langle \{\text{Lion}\}, \{\text{Pa, Af, la, sa}\} \rangle$,
$C_{11} = \langle \{\text{Panther}\}, \{\text{Pa, Af, As, me, sp, ye}\} \rangle$,
$C_{12} = \langle \{\text{Serval}\}, \{\text{Le, Af, sm, st, sp, sa, ye}\} \rangle$,
$C_{13} = \langle \{\text{Wildcat}\}, \{\text{Fe, Af, Am, As, Eu, sm, st, sp, bl, sa, ye}\} \rangle$,
$C_{14} = \langle \{\text{Cheetah}\}, \{\text{Ac, Af, me, st, sp, ye}\} \rangle$,
$C_{15} = \langle \emptyset, Y \rangle$.

The hierarchy of these clusters is depicted in Figure 2. As one can see, the new hierarchy is much easier to comprehend than the original one and it does not contain "artificial clusters" like the previous one. If we return to Figure 1, the black nodes represent clusters which are omitted in Figure 2 while the white ones represent clusters which are present in Figure 2. The hierarchy in Figure 2 contains two clusters which are trivial: C_1 (cluster of all animals) and C_{15} (cluster of no animals). These two borderline clusters may be omitted in the diagram.

One of the benefits of adding background knowledge to reduce the concept lattice is its interactive character. Namely, if user supplies AD-formulas and the structure is still too large, he/she can specify further restrictions which may help to further reduce the structure. For illustration, suppose we add the following AD-formula to the previous ones:

$$\{\text{Af, Am, As, Eu}\} \sqsubseteq \{\text{bl, sa, wh, ye}\}. \quad (6)$$

The AD-formula says that if a specification of a continent is present, then the specification of color must also be present in the concept description (intent). Using this AD-formula as an additional constraint, we arrive to 11 formal concepts as clusters. Each object (table row) induces one cluster—category of the corresponding species. In addition to that, there is cluster C_4 (cluster of all "*Panthera* felines"), which appears in the data and is compatible with the background knowledge. The situation is depicted in Figure 3. From Figure 3, we can see that some clusters such as C_6 (cluster of all "large *Panthera* felines from America") are no longer present in the hierarchy because they do not satisfy the constraint represented by (6).

Removing redundant AD-formulas can improve the readability of background knowledge. Background knowledge bases represented by AD-formulas may be collected from different sources, by different people, and during a longer period of time. In addition, if we take into account the number of attributes, which can be large, it is likely that the "same information" will be captured by different groups of rules in the knowledge base. This leads to a redundancy which is undesirable because it impairs comprehensibility of the background knowledge. We now demonstrate how Algorithm 2 can be used to remove redundant AD-formulas. As an example, consider background knowledge consisting of the following AD-formulas over attributes $a, \ldots, g$:

$$\{a, b\} \sqsubseteq \{c\}, \quad \{c, d\} \sqsubseteq \{a\}, \quad \{c, e\} \sqsubseteq \{a, d\},$$
$$\{f, g\} \sqsubseteq \{a, c\}, \quad \{d, f\} \sqsubseteq \{b, c\}, \quad \{g\} \sqsubseteq \{e, f\}.$$

The AD-formula $\{f, g\} \sqsubseteq \{a, c\}$ is redundant. In other words, $\{f, g\} \sqsubseteq \{a, c\}$ follows from the other AD-formulas. Following Algorithm 2, we take the complement of $\{a, c\}$ and compute the greatest model of the rest of the formulas which is smaller than or equal to the complement of $\{a, c\}$. Then, we conclude that $\{f, g\} \sqsubseteq \{a, c\}$ is entailed by the rest of the formulas, and therefore redundant, iff $\{f, g\}$ is disjoint with the computed model.

The complement of $\{a, c\}$ is $\{b, d, e, f, g\}$. Denote the complement by N. We now compute the greatest model according to Algorithm 1. Taking the first AD-formula $\{a, b\} \sqsubseteq \{c\}$, we see $b \in N \cap \{a, b\}$, but $c \notin N$, i.e. b is to be removed from N. The second formula $\{c, d\} \sqsubseteq \{a\}$ makes d removed from $N = \{d, e, f, g\}$ because $d \in N$ and $a \notin N$. The third formula $\{c, e\} \sqsubseteq \{a, d\}$ makes e removed from $N = \{e, f, g\}$ because $\{a, d\} \cap N = \emptyset$. The fourth formula is omitted because it is the formula for which we test redundancy. The fifth formula $\{d, f\} \sqsubseteq \{b, c\}$ makes f removed from $N = \{f, g\}$ because $f \in N$ and $\{b, c\} \cap N = \emptyset$. Finally, the last formula $\{g\} \sqsubseteq \{e, f\}$ causes g to be removed from $N = \{g\}$ because $g \in N$ and $\{e, f\} \cap N = \emptyset$. Therefore, we have arrived to $N = \emptyset$, i.e. $N \cap \{f, g\} = \emptyset$, which, according to Theorem 2 means that $\{f, g\} \sqsubseteq \{a, c\}$ follows from the other AD-formulas.

4. REMARKS AND FUTURE RESEARCH

We showed that AD-formulas provide us with an easy-to-understand and theoretically and computationally tractable way to add background knowledge into FCA. The approach is relationally based and thus avoids *ad-hoc* assignment of weights to attributes which is an approach usually used for modeling attribute importance.

Due to limited scope, we presented only illustrative examples. Two of our ongoing projects where background knowledge via AD-formulas plays important role are: 1. Development of a conceptual clustering system of machine parts. 2. Development of a system for on-line shops which provides the customer with a conceptual view of the products, grouped into categories, based on user preferences which play the role of background knowledge. Experience with these projects supports our claim that adding background knowledge to FCA yields a flexible and user-friendly categorization tool.

Future research will focus on further methodological and theoretical development of using background knowledge in FCA as well as on real-world applications of FCA with background knowledge.

5. ACKNOWLEDGMENTS

The authors' second affiliation is with Dept. Computer Science, Palacky University, Olomouc, Czech Republic. Supported by grant No. 1ET101370417 of GA AV ČR, by grant No. 201/05/0079 of the Czech Science Foundation, and by institutional support, research plan MSM 6198959214.

6. REFERENCES

[1] Belohlavek R., Sklenář V.: Formal concept analysis constrained by attribute-dependency formulas. ICFCA 2005, LNCS **3403**, pp. 176–191, 2005.

[2] Belohlavek R., Vychodil V.: Semantic entailment of attribute-dependency formulas and their non-redundant bases. Proc. JCIS 2006, Joint Conference on Information Sciences, Kaohsiung, Taiwan, ROC, pp. 747–750.

[3] Carpineto C., Romano G.: *Concept Data Analysis. Theory and Applications.* J. Wiley, 2004.

[4] Davidson I., Ravi S. S.: Hierarchical clustering with constraints: theory and practice. PKDD 2005, LNAI **3721**, pp. 59–70.

[5] Diognon J.-P., Falmagne J.-C.: *Knowledge Spaces.* Springer, 1999.

[6] Ganter B.: Attribute exploration with background knowledge. *Theor. Comput. Sci.* **217**(1999), 215–233.

[7] Ganter B., Wille R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, 1999.

[8] Ganter B., Wille R.: Contextual attribute logic. In: Tepfenhart W., Cyre W. (Eds.): *Proceedings of ICCS 2001*, Springer, 2001.

[9] Lloyd, J. W.: *Foundations of Logic Programming* (2nd ed.). Springer-Verlag, New York, 1987.

[10] Maier D.: *The Theory of Relational Databases.* Computer Science Press, Rockville, 1983.

[11] Stumme G.: Attribute Exploration with Background Implications and Exceptions. In H.-H. Bock and W. Polasek (Eds.): Data Analysis and Information Systems. Statistical and Conceptual approaches. Data Analysis, and Knowledge Organization 7, Springer, 1996, pp. 457–469.

[12] Wagsta K., Cardie C., Rogers S., Schroedl S.: Constrained K-means Clustering with Background Knowledge. ICML 2001, Williamstown, MA, pp. 577–584.